TOPICS IN BIVARIATE SPLINE SMOOTHING

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Luo Xiao

August 2012

TOPICS IN BIVARIATE SPLINE SMOOTHING

Luo Xiao, Ph.D.

Cornell University 2012

Penalized spline methods have been popular since the work of Eilers and Marx (1996). Recent years saw extensive theoretical studies and a wide range of applications of penalized splines. In this dissertation, we consider penalized splines for smoothing two-dimensional data.

In Chapter 2, we propose a new spline smoother, *the sandwich smoother*, for smoothing data on a rectangular grid. Univariate P-spline smoothers are applied simultaneously along both coordinates. The sandwich smoother has a tensor product structure that simplifies an asymptotic analysis and it can be fast computed. We derive a local central limit theorem for the sandwich smoother, with simple expressions for the asymptotic bias and variance, by showing that the sandwich smoother is asymptotically equivalent to a bivariate kernel regression estimator with a product kernel. As far as we are aware, this is the first central limit theorem for a bivariate spline estimator of any type. Our simulation study shows that the sandwich smoother is orders of magnitude faster to compute than other bivariate spline smoothers, even when the latter are computed using a fast GLAM (Generalized Linear Array Model) algorithm, and comparable to them in terms of mean squared integrated errors. One important application of the sandwich smoother is to estimate covariance functions in functional data analysis. In this application, our numerical results show that the sandwich smoother is orders of magnitude faster than local linear regression.

In Chapter 3, based on the sandwich smoother, we propose a fast covariance

function estimation method (FACE) for smoothing high-dimensional functional data. We show that our method overcomes the computational difficulty of common bivariate smoothers for smoothing high-dimensional covariance operators, and in particular we derive a fast algorithm for selecting the smoothing parameter. We also show that through FACE we can simultaneously obtain the smoothed covariance operator and its associated eigenfunctions. For functional principal component analysis, we derive a fast method for calculating the principal scores. A simulation study is done to illustrate the computational speed of FACE.

Although not a focus of this dissertation, we present in Appendix A a theoretical study of the local asymptotics of P-splines for the univariate case. In this work we derived the local asymptotic distribution of P-splines at both an interior point and near the boundary. Some of the results in the work are used in studying the sandwich smoother.

BIOGRAPHICAL SKETCH

Luo Xiao was born in Xixian, Henan Province, China, on January 12, 1983. He went to Xinyang Senior Middle School in 1998 and graduated in 2001. He studied in University of Science and Technology of China since August, 2001 and received a degree in Bachelor of Science in Mathematics in 2005. In the fall of 2005, he went to University of Pennsylvania and received a degree in Master of Arts in Mathematics in 2007. After that he came to Cornell University to pursue his Ph.D. in the field of statistics. In 2011, he married Ms. Heli Chen.

To my dear wife Heli Chen and our parents

ACKNOWLEDGEMENTS

I would like to express my sincere and deep gratitude to my advisor, Professor David Ruppert, who has guided me throughout my years at Cornell. He taught me to be a good researcher, writer, and speaker. I am grateful to Dr. Ruppert for his patience and time spent on helping me work out research problems. He is a brilliant and dedicated mentor and I learnt a lot from him.

I would also like to thank my thesis committee members, Drs. Giles Hooker and Robert L. Strawderman, for their invaluable guidance and help in the work and beyond. I would like to thank both for their generosity in providing financial support.

I am grateful to my parents, brother, and sister, for their endless support over the years. Many thanks go to my lifelong friend, Mr. Zhe Nan, for his friendship over the last ten years. I also want to thank Dr. Xin Ma for her help and encouragement in the past year.

Finally, I would like to express my appreciation and thanks to my dear wife, Heli Chen, for her love, care, understanding, support and encouragement. I am fortunate to have married her and she is a wonderful, lovely, and supportive wife.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In regression models it is often difficult to assume specific forms for the regression functions and hence it is preferable to make few assumptions about these functions. Nonparametric regressions, or smoothing methods, provide powerful tools to model these functions. Common smoothing methods include kernel (Nadaraya, 1964; Watson, 1964; Gasser and Müller, 1979), local polynomial (see, e.g., Fan, 1992), smoothing splines (see, e.g., Wahba, 1990; Gu, 2002) and penalized splines (see, e.g., Eilers and Marx, 1996; Ruppert *et al.*, 2003).

A spline is a piecewise polynomial function that are smoothly connected at the changepoints. The changepoints are called knots. Splines were first used for interpolation in numerical analysis. Spline interpolation may be preferred to polynomial approximation because it yields a similar result while avoiding the oscillation between data points when polynomials of high degrees are used in the approximation. Other useful properties of splines have also been found, such as stability of evaluation and capacity to approximate curves with complex structures. See de Boor (2001) for a comprehensive study of splines.

The extensive research of Grace Wahba as well as other researchers demonstrate that smoothing splines provide flexible data analysis tools for a wide range of statistical problems. It is worth to mention that an efficient algorithm by Hutchinson and de Hoog (1995) contributes to the popularity of smoothing splines in statistical analysis. Thin plate splines, i.e., smoothing splines for more than one dimension, however, is generally computationally expensive as about n^3 computations are needed for n data points. To reduce the computational complexity of thin plate splines, Wood (2003) proposed thin plate regression splines as an approximation

to thin plate splines. Wahba (1990) and Gu (2002) are two excellent monographs on smoothing splines.

Penalized splines (see, e.g., Eilers and Marx, 1996) use a smaller number of spline bases to approximate the regression function and control the smoothness of the approximation with a penalty similar to that of smoothing splines. So penalized splines need less computation than smoothing splines. See Ruppert *et al.* (2003) or Wood (2006) for both methodological development and applications. For smoothing of two continuous variables, Eilers and Marx (2003) proposed a bivariate P-spline method which is easier to compute than thin plate regression splines.

It is also of interest to study the asymptotics of the spline methods. The asymptotic normality of smoothing splines was established in Silverman (1984) and the convergence rate of smoothing splines in general contexts can be found in Gu (2002). The theoretical study of penalized splines, however, has been challenging. An asymptotic study of univariate penalized splines was achieved only recently. To be specific, Opsomer and Hall (2005) first studied the asymptotic theory of penalized splines when the number of knots is infinite. Li and Ruppert (2008) derived the first asymptotic distribution of penalized splines when splines of low degrees and a penalty of low order are used. Wang *et al.* (2011) connected penalized splines with some ordinary differential equations (ODEs), and by studying Greens functions associated with those ODEs, they were able to derive the asymptotic distribution of penalized splines. In contrast to Li and Ruppert (2008), Kauermann *et al.* (2009) considered the situation when the number of knots increases at a slower rate. Though they did not obtain an explicit expression for the asymptotic bias and variance, they generalized their results for non-normal responses. Claeskens *et*

al. (2009) showed that depending on whether the number of knots increasing at a sufficiently fast or a sufficiently slow rate, the asymptotic distribution of penalized splines is either close to that of a smoothing spline or to a regression spline. As a consequence, they referred to these two cases as either a large or small number of knots scenario. The large number of knots scenario is closer to current practice, as discussed, for example, in O' Sullivan (1986), Eilers and Marx (1996), and Ruppert *et al.* (2003), a relatively large number of knots is used and overfitting is offset by the penalty with an appropriate smoothing parameter.

Spline methods for bivariate smoothing have not been as thoroughly studied as for the univariate case. For instance, no theory has been established for thin plate splines or the bivariate P-spline method by Eilers and Marx (2003). Besides absence of theory, bivariate spline smoothing can be computationally hard. As mentioned above, to fit thin plate splines to n data points requires about n^3 computations. For the same problem, the thin plate regression splines requires about kn^2 computations where k is the basis dimension. For the bivariate P-spline method by Eilers and Marx (2003) the main computational burden is on selecting the two smoothing parameters which can be computationally expensive when a large number of knots are used. When the data is collected from a rectangular grid, the generalized linear array model (GLAM) by Currie *et al.* (2006) provides a low storage, high speed algorithm by making use of the matrix structures of the model matrix and the data. The bivariate P-splines can be sped up when implemented with a GLAM algorithm although there is still no fast selection of smoothing parameters.

An important example of bivariate smoothing is covariance function estimation. Covariance function is an important part of functional and longitudinal data

analysis and in the literature common bivariate smoothers such as kernel, local polynomial and penalized splines have been used to estimate covariance functions (Staniswalis and Lee, 1998; Yao *et al.*, 2005; Yao and Lee, 2006; Di *et al.*, 2009). In functional data analysis, functional data are usually densely measured with a large number of data points. Hence covariance function estimation by these smoothers can be computationally hard and even infeasible if the number of measurements per subject exceeds, say, 500. Often the eigenfunctions of the covariance function are of interest and are in practice approximated by the eigendecomposition of the discretized covariance function. When the matrix formed by discretizing the covariance function is of large size, its eigendecomposition is computationally nontrivial. Hence it is of practical interest to build a bivariate smoothing method that could directly estimate the eigenfunctions of the covariance function.

In the following chapters we study a new bivariate spline method for smoothing data on a rectangular grid. Aiming at smoothing grid data of large size, the new method will be more computationally advantageous than existing methods if it can be implemented with a GLAM algorithm and also has a fast selection of smoothing parameters. The new method can be quite useful for functional data analysis if it can simultaneously estimate the covariance function and the associated eigenfunctions.

Although not a focus of this dissertation, appendix A presents some of my theoretic work on univariate P-splines, joint with other researchers. In this work, we derived the local asymptotic distribution of P-splines at both an interior point and near the boundary. We showed that the convergence rate of P-splines near the boundary is slower than at an interior point. The method used in this work is different from Wang *et al.* (2009).

CHAPTER 2

FAST BIVARIATE P-SPLINES: THE SANDWICH SMOOTHER

2.1 Introduction

This chapter is based on joint work with Yingxing Li and David Ruppert.

A bivariate smoother is a function or procedure for drawing a smooth surface of two continuous variables. Being nonparametric, bivariate smoothers allow for complicated interactions of the two variables and hence are useful alternatives to multivariate linear or parametric models. Many bivariate smoothers are natural extensions of univariate smoothers such as kernel (Nadaraya, 1964; Watson, 1964; Gasser and Müller, 1979) and local polynomial (Fan, 1992).

Spline is an important smoothing method. For bivariate spline smoothing, there are two well known estimators: bivariate P-splines (Eilers and Marx, 2003; Marx and Eilers, 2005) and thin plate splines, e.g., the thin plate regression splines (Wood, 2003). For convenience, the Eilers-Marx and Wood estimators will be denoted by E-M and TPRS, respectively. We use E-M without specification of how the estimator is calculated.

In this work, we propose a fast penalized spline method for bivariate smoothing. Univariate P-spline smoothing (Eilers and Marx, 1996) is applied simultaneously along both coordinates. The new smoother is named the sandwich smoother as it can be written in a sandwich form. The sandwich smoother has a tensor product structure that simplifies the asymptotic analysis and also facilitates a fast algorithm for generalized cross validation. We derive a local central limit theorem for the sandwich smoother, with simple expressions for the asymptotic bias and variance,

by showing that the sandwich smoother is asymptotically equivalent to a bivariate kernel regression estimator with a product kernel.

Specifically, in Section 2.2 we provide details about the sandwich smoother. In Section 2.3, we establish an asymptotic theory of the sandwich smoother and show that it is asymptotically equivalent to a bivariate kernel estimator using a product kernel. In Section 2.4, we consider irregularly spaced data. In Section 2.5, we report a simulation study. In Section 2.6, we study the sandwich smoother for estimating covariance functions through a simulation study and then we apply the method to some medical imaging data that is publicly available. In Section 2.7, we extend the fast bivariate P-spline to the multivariate P-spline which handles array data of dimensions greater than two.

2.2 The Sandwich Smoother

Suppose there is a bivariate regression function $\mu(x, z)$ with $(x, z) \in [0, 1]^2$. The model is $y_{i,j} = \mu(x_i, z_j) + \epsilon_{i,j}, 1 \leq i \leq n_1, 1 \leq j \leq n_2$, where the $\epsilon_{i,j}$'s are independent with $\mathrm{E}\epsilon_{i,j} = 0$ and $\mathrm{E}\epsilon_{i,j}^2 = \sigma^2(x_i, z_j)$, the design points $\{(x_i, z_j)\}_{1 \leq i \leq n_1, 1 \leq j \leq n_2}$ are deterministic, and the total number of data points is $n = n_1 n_2$. The cases with fixed design points not in a regular grid or with random design points are studied in Section 2.4. Then the data can be organized into $\mathbf{Y}$, a matrix of dimension $n_1 \times n_2$. We propose to smooth across the rows of the grid and down the columns of the data matrix $\mathbf{Y}$ so that the matrix of fitted values $\hat{\mathbf{Y}}$ satisfies

$$\hat{\mathbf{Y}} = \mathbf{S}_1 \mathbf{Y} \mathbf{S}_2, \tag{2.1}$$

where $\mathbf{S}_1$ (or $\mathbf{S}_2$) is the smoother matrix for x (or z) in (2.3). So fixing one covariate, we smooth along the other covariate and vice versa, although the two smooths are

simultaneous as implied by (2.1). The metaphor of the name "sandwich smoother" can be seen in equation (2.1): the two smoother matrices are two pieces of breads and the data matrix is a piece of ham in between the breads. This is similar to the sandwich form of the covariance matrix in generalized linear models. The sandwich smoother can be quite useful because it can computed by a fast algorithm derived in Section 2.2.2. After we have finished this work, we learn that Dierckx (1982) proposed a smoother with the same structure as (2.1). However the asymptotic analysis and the fast algorithm for the sandwich smoother are new. We also learn that for smoothing two-dimensional histograms, Eilers and Goeman (2004) studied a simplified version of the sandwich smoother with special smoother matrices that lead to non-negative smooth for non-negative data. Hence the fast algorithm for the sandwich smoother can be applied to their method.

Let vec be the operation that stacks the columns of a matrix into a vector. Define $\mathbf{y} = \mathrm{v}ec(\mathbf{Y})$ and $\hat{\mathbf{y}} = \mathrm{v}ec(\hat{\mathbf{Y}})$. Applying a well-known identity of the tensor product (Lemma 2.1) to (2.1) gives

$$\hat{\mathbf{y}} = (\mathbf{S}_2 \otimes \mathbf{S}_1)\mathbf{y}. \tag{2.2}$$

Identity (2.2) shows that the overall smoother matrix is a tensor product of two univariate smoother matrices. Because of this factorization of the smoother matrix, we say our model has a tensor product structure. We use P-splines (Eilers and Marx, 1996) to construct univariate smoother matrices, i.e.,

$$\mathbf{S}_i = \mathbf{B}_i(\mathbf{B}_i^T\mathbf{B}_i + \lambda_i\mathbf{D}_i^T\mathbf{D}_i)^{-1}\mathbf{B}_i^T, i = 1, 2, \tag{2.3}$$

where $\mathbf{B}_1$ and $\mathbf{B}_2$ are the model matrices for x and z using B-spline basis (defined later), and $\mathbf{D}_1$ and $\mathbf{D}_2$ are differencing matrices of difference orders m_1 and m_2, respectively. Then the overall smoother matrix can be written out using identities

of the tensor product (Lemma 2.2),

$$\begin{aligned}\mathbf{S}_2 \otimes \mathbf{S}_1 &= \left\{\mathbf{B}_2(\mathbf{B}_2^T\mathbf{B}_2 + \lambda_2\mathbf{D}_2^T\mathbf{D}_2)^{-1}\mathbf{B}_2^T\right\} \otimes \left\{\mathbf{B}_1(\mathbf{B}_1^T\mathbf{B}_1 + \lambda_1\mathbf{D}_1^T\mathbf{D}_1)^{-1}\mathbf{B}_1^T\right\} \\ &= (\mathbf{B}_2 \otimes \mathbf{B}_1)\{\mathbf{B}_2^T\mathbf{B}_2 \otimes \mathbf{B}_1^T\mathbf{B}_1 + \lambda_1\mathbf{B}_2^T\mathbf{B}_2 \otimes \mathbf{D}_1^T\mathbf{D}_1 \\ &\quad + \lambda_2\mathbf{D}_2^T\mathbf{D}_2 \otimes \mathbf{B}_1^T\mathbf{B}_1 + \lambda_1\lambda_2\mathbf{D}_2^T\mathbf{D}_2 \otimes \mathbf{D}_1^T\mathbf{D}_1\}^{-1}(\mathbf{B}_2 \otimes \mathbf{B}_1)^T. \end{aligned} \tag{2.4}$$

The inverse matrix in the second equality of (2.4) shows that our model uses tensor product splines (defined later) with penalty

$$\mathbf{P} = \lambda_1\mathbf{B}_2^T\mathbf{B}_2 \otimes \mathbf{D}_1^T\mathbf{D}_1 + \lambda_2\mathbf{D}_2^T\mathbf{D}_2 \otimes \mathbf{B}_1^T\mathbf{B}_1 + \lambda_1\lambda_2\mathbf{D}_2^T\mathbf{D}_2 \otimes \mathbf{D}_1^T\mathbf{D}_1 \tag{2.5}$$

on the coefficients matrix (defined below). The tensor product splines of two variables (Dierckx 1995, ch. 2) is defined by

$$\sum_{1\le\kappa\le c_1, 1\le\ell\le c_2} \theta_{\kappa,\ell} B_\kappa^1(x) B_\ell^2(z),$$

where B_κ^1 and B_ℓ^2 are B-spline basis functions for x and z, respectively, c_1 and c_2 are the numbers of basis functions for the univariate splines, and $\mathbf{\Theta} = (\theta_{\kappa,\ell})_{1\le\kappa\le c_1, 1\le\ell\le c_2}$ is the coefficients matrix. We use B-splines of degrees p_1 (p_2) for x (z), and use $K_1 - 1$ ($K_2 - 1$) equidistant interior knots. Then $c_1 = K_1 + p_1$, $c_2 = K_2 + p_2$. It follows that the model is

$$\mathbf{Y} = \mathbf{B}_1\mathbf{\Theta}\mathbf{B}_2^T + \boldsymbol{\epsilon}, \tag{2.6}$$

where $\mathbf{B}_1 = \{B_\kappa^1(x_r)\}_{1\le r\le n_1, 1\le\kappa\le c_1}$, $\mathbf{B}_2 = \{B_\ell^2(z_s)\}_{1\le s\le n_2, 1\le\ell\le c_2}$, and $\boldsymbol{\epsilon}$ is an $n_1 \times n_2$ matrix with $(i,j)th$ entry $\epsilon_{i,j}$. Let $\boldsymbol{\theta}$ =vec($\mathbf{\Theta}$). Then an estimate of $\boldsymbol{\theta}$ is given by minimizing $\|\mathbf{Y} - \mathbf{B}_1\hat{\mathbf{\Theta}}\mathbf{B}_2^T\|_F^2 + \hat{\boldsymbol{\theta}}^T\mathbf{P}\hat{\boldsymbol{\theta}}$, where $\|\cdot\|_F$ is the Frobenius norm and $\mathbf{P}$ is defined in (2.5). It follows that the estimate of the coefficients matrix $\hat{\mathbf{\Theta}}$ satisfies $\mathbf{\Lambda}_1\hat{\mathbf{\Theta}}\mathbf{\Lambda}_2 = \mathbf{B}_1^T\mathbf{Y}\mathbf{B}_2$, where for $i = 1, 2$, $\mathbf{\Lambda}_i = \mathbf{B}_i^T\mathbf{B}_i + \lambda_i\mathbf{D}_i^T\mathbf{D}_i$, or equivalently, $\hat{\boldsymbol{\theta}}$ satisfies

$$(\mathbf{\Lambda}_2 \otimes \mathbf{\Lambda}_1)\,\hat{\boldsymbol{\theta}} = (\mathbf{B}_2 \otimes \mathbf{B}_1)^T\mathbf{y}. \tag{2.7}$$

Then our penalized estimate is

$$\hat{\mu}(x,z) = \sum_{1\le\kappa\le c_1, 1\le\ell\le c_2} \hat{\theta}_{\kappa,\ell} B^1_\kappa(x) B^2_\ell(z). \tag{2.8}$$

With (2.7), it is straightforward to show that $\hat{\mathbf{y}} = (\mathbf{B}_2 \otimes \mathbf{B}_1)\hat{\boldsymbol{\theta}}$ satisfies (2.1), which confirms that the proposed method uses tensor product splines with a particular penalty.

2.2.1 Comparison with the E-M estimator

The only difference between the sandwich smoother and the E-M estimator (Marx and Eilers, 2003; Eilers and Marx, 2006) is the penalty. Let $\mathbf{P}_{\text{E-M}}$ denote the penalty matrix for the E-M estimator, then $\mathbf{P}_{\text{E-M}} = \lambda_1 \mathbf{I}_{c_2} \otimes \mathbf{D}_1^T\mathbf{D}_1 + \lambda_2 \mathbf{D}_2^T\mathbf{D}_2 \otimes \mathbf{I}_{c_1}$, where $\mathbf{I}_{c_i}$ is an identity matrix of dimension c_i. The first and second penalty terms in bivariate P-splines penalize the columns and rows of $\boldsymbol{\Theta}$, respectively, and are thus called column and row penalties. It can be shown that the first penalty term in (2.5), $\mathbf{B}_2^T\mathbf{B}_2 \otimes \mathbf{D}_1^T\mathbf{D}_1$, like $\mathbf{I}_{c_2} \otimes \mathbf{D}_1^T\mathbf{D}_1$, is a "column" penalty, but it penalizes the columns of $\boldsymbol{\Theta}\mathbf{B}_2^T$ instead of the columns of $\boldsymbol{\Theta}$. We call this a modified column penalty. The implication of this modified column penalty can be seen from a closer look at model (2.6). By regarding (2.6) as a model with B-spline base $\mathbf{B}_1$ and coefficients $\boldsymbol{\Theta}\mathbf{B}_2^T$, (2.6) becomes a varying-coefficients model (Hastie and Tibshirani, 1993) in x with coefficients depending on z. So we can interpret the modified column penalty as a penalty for the univariate P-spline smoothing along the x-axis. Similarly, the penalty term $\mathbf{D}_2^T\mathbf{D}_2 \otimes \mathbf{B}_1^T\mathbf{B}_1$ for the sandwich smoother penalizes the rows of $\mathbf{B}_1\boldsymbol{\Theta}$ and can be interpreted as the penalty for the univariate P-spline smoothing along the z-axis. The third penalty in (2.4) corresponds to the interaction of the two univariate smoothing.

2.2.2 Fast Algorithm

We derive a fast algorithm for the sandwich smoother by showing how the smoothing parameters can be selected via a fast computation of GCV. GCV requires the computation of $\|\hat{\mathbf{Y}} - \mathbf{Y}\|_F^2$ and the trace of the overall smoother matrix. We need some initial computations. First, we need the singular value decompositions

$$(\mathbf{B}_i^T\mathbf{B}_i)^{-1/2}\mathbf{D}_i^T\mathbf{D}_i(\mathbf{B}_i^T\mathbf{B}_i)^{-1/2} = \mathbf{U}_i\text{diag}(\mathbf{s}_i)\mathbf{U}_i^T, \quad \text{for } i = 1, 2, \tag{2.9}$$

where $\mathbf{U}_i$ is the matrix of eigenvectors and $\mathbf{s}_i$ is the vector of eigenvalues. For $i = 1, 2$, let $\mathbf{A}_i = \mathbf{B}_i(\mathbf{B}_i^T\mathbf{B}_i)^{-1/2}\mathbf{U}_i$, then $\mathbf{A}_i^T\mathbf{A}_i = \mathbf{I}_{c_i}$ and $\mathbf{A}_i\mathbf{A}_i^T = \mathbf{B}_i(\mathbf{B}_i^T\mathbf{B}_i)^{-1}\mathbf{B}_i^T$. It follows that for $i = 1, 2$, $\mathbf{S}_i = \mathbf{A}_i\mathbf{\Sigma}_i\mathbf{A}_i^T$ with $\mathbf{\Sigma}_i = \{\mathbf{I}_{c_i} + \lambda_i\text{diag}(\mathbf{s}_i)\}^{-1}$.

We first compute $\|\hat{\mathbf{Y}} - \mathbf{Y}\|_F^2$. Substituting $\mathbf{A}_i\mathbf{\Sigma}_i\mathbf{A}_i^T$ for $\mathbf{S}_i$ in equation (2.1) we obtain

$$\hat{\mathbf{Y}} = \mathbf{A}_1\left\{\mathbf{\Sigma_1}\left(\mathbf{A}_1^T\mathbf{Y}\mathbf{A}_2\right)\mathbf{\Sigma_2}\right\}\mathbf{A}_2^T = \mathbf{A}_1\left(\mathbf{\Sigma_1}\tilde{\mathbf{Y}}\mathbf{\Sigma_2}\right)\mathbf{A}_2^T,$$

where $\tilde{\mathbf{Y}} = \mathbf{A}_1^T\mathbf{Y}\mathbf{A}_2$. Let $\tilde{\mathbf{y}} = vec(\tilde{\mathbf{Y}})$, then

$$\hat{\mathbf{y}} = (\mathbf{A}_2 \otimes \mathbf{A}_1)(\mathbf{\Sigma}_2 \otimes \mathbf{\Sigma}_1)\tilde{\mathbf{y}}. \tag{2.10}$$

We shall use the following operations on vectors: let $\mathbf{a}$ be a vector containing only positive elements, $\mathbf{a}^{1/2}$ denotes the element-wise squared root of $\mathbf{a}$ and $1/\mathbf{a}$ denotes the element-wise inverses of $\mathbf{a}$. First we have

$$\|\hat{\mathbf{Y}} - \mathbf{Y}\|_F^2 = (\hat{\mathbf{y}} - \mathbf{y})^T(\hat{\mathbf{y}} - \mathbf{y}) = \hat{\mathbf{y}}^T\hat{\mathbf{y}} - 2\hat{\mathbf{y}}^T\mathbf{y} + \mathbf{y}^T\mathbf{y}.$$

It can be shown by (2.10) that

$$\begin{aligned}
\hat{\mathbf{y}}^T\hat{\mathbf{y}} &= \tilde{\mathbf{y}}^T(\mathbf{\Sigma}_2 \otimes \mathbf{\Sigma}_1)(\mathbf{A}_2 \otimes \mathbf{A}_1)^T(\mathbf{A}_2 \otimes \mathbf{A}_1)(\mathbf{\Sigma}_2 \otimes \mathbf{\Sigma}_1)\tilde{\mathbf{y}} \\
&= \tilde{\mathbf{y}}^T(\mathbf{\Sigma}_2 \otimes \mathbf{\Sigma}_1)(\mathbf{\Sigma}_2 \otimes \mathbf{\Sigma}_1)\tilde{\mathbf{y}} \\
&= |\tilde{\mathbf{y}}^T(\mathbf{\Sigma}_2 \otimes \mathbf{\Sigma}_1)|^2 \\
&= \left\{\tilde{\mathbf{y}}^T(\tilde{\mathbf{s}}_2 \otimes \tilde{\mathbf{s}}_1)\right\}^2,
\end{aligned}$$

where $\tilde{\mathbf{s}}_i = 1/(\mathbf{1}_{c_i} + \lambda_i \mathbf{s}_i)$ for $i = 1, 2$ and $\mathbf{1}_{c_i}$ is a vector of 1's with length c_i. In the above derivation, $|\cdot|$ denotes the Euclidean norm in the second to last equality; we used the fact that $\mathbf{A}_i^T \mathbf{A}_i = \mathbf{I}_{c_i}$ and also both $\mathbf{\Sigma}_2$ and $\mathbf{\Sigma}_1$ are diagonal matrices. Similarly we obtain

$$\hat{\mathbf{y}}^T \mathbf{y} = \left\{ \tilde{\mathbf{y}}^T \left(\tilde{\mathbf{s}}_2^{1/2} \otimes \tilde{\mathbf{s}}_1^{1/2} \right) \right\}^2.$$

It follows that

$$\|\hat{\mathbf{Y}} - \mathbf{Y}\|_F^2 = \left\{ \tilde{\mathbf{y}}^T \left(\tilde{\mathbf{s}}_2 \otimes \tilde{\mathbf{s}}_1 \right) \right\}^2 - 2 \left\{ \tilde{\mathbf{y}}^T \left(\tilde{\mathbf{s}}_2^{1/2} \otimes \tilde{\mathbf{s}}_1^{1/2} \right) \right\}^2 + \mathbf{y}^T \mathbf{y}. \tag{2.11}$$

The right hand of (2.11) shows that for each pair of smoothing parameters the calculation of $\|\hat{\mathbf{Y}} - \mathbf{Y}\|_F^2$ is just two inner product of vectors of length $c_2 c_1$ and the term $\mathbf{y}^T \mathbf{y}$ just needs one calculation for all smoothing parameters.

Next, the trace of the overall smoother matrix can be computed by first using another identity of the tensor product (Lemma 2.3)

$$tr(\mathbf{S}_2 \otimes \mathbf{S}_1) = tr(\mathbf{S}_2) \cdot tr(\mathbf{S}_1), \tag{2.12}$$

and then using a trace identity (Seber, 2007, pp. 55) and as well as the fact that $\mathbf{A}_i^T \mathbf{A}_i = \mathbf{I}_{c_i}$,

$$tr(\mathbf{S}_i) = \sum_{\kappa=1}^{c_i} \frac{1}{1 + \lambda_i s_{i,\kappa}}, \tag{2.13}$$

where $s_{i,\kappa}$ is the κth element of $\mathbf{s}_i$.

By equations (2.11), (2.12) and (2.13) we obtain a fast algorithm for computing GCV that enables us to select the smoothing parameters efficiently:

A fast algorithm for selecting the smoothing parameters.

(1) Compute the singular value decompositions in (2.9).

(2) Compute $\mathbf{A}_i = \mathbf{B}_i (\mathbf{B}_i^T \mathbf{B}_i)^{-1/2} \mathbf{U}_i$, i =1, 2.

(3) Compute $\tilde{\mathbf{Y}} = \mathbf{A}_1^T \mathbf{Y} \mathbf{A}_2$ and $\tilde{\mathbf{y}} = vec(\tilde{\mathbf{Y}})$.

(4) For every pair of smoothing parameters

(a) Compute $\|\hat{\mathbf{Y}} - \mathbf{Y}\|_F^2$ by (2.11).

(b) Compute $tr(\mathbf{S})$ by (2.12) and (2.13).

(c) Compute $GCV = \|\hat{\mathbf{Y}} - \mathbf{Y}\|_F^2/(1 - tr(\mathbf{S})/n)^2$.

(5) Select the pair of smoothing parameters with the smallest GCV.

Because of the above fast algorithm, the sandwich smoother can be much faster than bivariate P-splines implemented with a GLAM algorithm; see Section 2.5.2 for an empirical comparison. For the E-M estimator, the inverse of a matrix of dimension $c_1 c_2 \times c_1 c_2$ is required for every pair of (λ_1, λ_2), while for the sandwich smoother, except in the initial computations in (2.9), no matrix inversion is required.

2.3 Asymptotic Theory

In this section, we derive the asymptotic distribution of the sandwich smoother and show that it is asymptotically equivalent to a bivariate kernel regression estimator with a product kernel. Moreover, we show that when the two orders of difference penalties are the same, our estimator has the optimal rate of convergence.

We shall use the equivalent kernel method first used for studying smoothing splines (Silverman, 1984) and also useful in studying the asymptotics of P-splines (Li and Ruppert, 2008; Wang *et al.*, 2011). A nonparametric point estimate is usually a weighted average of all data points, with the weights depending on the point

and the method being used. The equivalent kernel method shows that the weights are asymptotically the weights from a kernel regression estimator for some kernel function (the equivalent kernel) and some bandwidth (the equivalent bandwidth). First, we define a univariate kernel function

$$H_m(x) = \sum_{\nu=1}^{m} \frac{\psi_\nu}{2m} \exp\{-\psi_\nu |x|\}, \tag{2.14}$$

where m is a positive integer and the ψ_ν's are the m complex roots of $x^{2m}+(-1)^m = 0$ that have positive real parts. Here H_m is the equivalent kernel for univariate penalized splines (Wang *et al.*, 2011). By Lemma A.13 in Section A.9 of Appendix A, H_m is of order $2m$. Note that the order of a kernel determines the convergence rate of the kernel estimator. See Wand and Jones (1995) for more details. A bivariate kernel regression estimator with the product kernel $H_{m_1}(x)H_{m_2}(z)$ is of the form $(nh_{n,1}h_{n,2})^{-1}\sum_{i,j} y_{i,j} H_{m_1}\left\{h_{n,1}^{-1}(x-x_i)\right\} H_{m_2}\left\{h_{n,2}^{-1}(z-z_j)\right\}$, where $h_{n,1}$ and $h_{n,2}$ are the bandwidths. Under appropriate assumptions, the sandwich smoother is asymptotically equivalent to the above kernel estimator (Proposition 2.1). Because the asymptotic theory of a kernel regression estimator is well established (Wand and Jones, 1995), an asymptotic theory can be similarly established for the sandwich smoother. For notational convenience, $a \sim b$ implies a/b converges to 1.

Proposition 2.1 *Assume the following conditions are satisfied.*

1. *There exists a constant* $\delta > 0$ *such that* $\sup_{i,j} \mathrm{E}\left(|y_{i,j}|^{2+\delta}\right) < \infty$.
2. *The regression function* $\mu(x, z)$ *has continuous* $2m$*th order derivatives where* $m = \max(m_1, m_2)$.
3. *The variance function* $\sigma^2(x, z)$ *is continuous.*
4. *The covariates satisfy* $(x_i, z_j) = ((i - 1/2)/n_1, (j - 1/2)/n_2)$.

5. $n_1 \sim c_n n_2$ *where* c_n *is a constant.*

Let $h_{n,1} = K_1^{-1}(\lambda_1 K_1 n_1^{-1})^{1/(2m_1)}$, $h_{n,2} = K_2^{-1}(\lambda_2 K_2 n_2^{-1})^{1/(2m_2)}$ *and* $h_n = h_{n,1}h_{n,2}$. *Assume* $h_{n,1} = O(n^{-\nu_1})$ *and* $h_{n,2} = O(n^{-\nu_2})$ *for some constants* $0 < \nu_1, \nu_2 < 1$. *Assume also* $(K_1 h_{n,1}^2)^{-1} = o(1)$ *and* $(K_2 h_{n,2}^2)^{-1} = o(1)$. *Let* $\hat{\mu}(x,z)$ *be the sandwich smoother using* m_1*th* (m_2*th*) *order difference penalty and* $p_1 \geq 1$ ($p_2 \geq 1$) *degree B-splines on the x-axis (z-axis) with equally spaced knots. Fix* $(x,z) \in (0,1) \times (0,1)$. *Let* $\mu^*(x,z) = (nh_n)^{-1} \sum_{i,j} y_{i,j} H_{m_1}\left\{h_{n,1}^{-1}(x - x_i)\right\} H_{m_2}\left\{h_{n,2}^{-1}(z - z_j)\right\}$. *Then*

$$\begin{aligned} \mathrm{E}\left\{\hat{\mu}(x,z) - \mu^*(x,z)\right\} &= O\left[\max\{(K_1 h_{n,1})^{-2}, (K_2 h_{n,2})^{-2}\}\right], \\ \mathrm{var}\{\hat{\mu}(x,z) - \mu^*(x,z)\} &= o\{(nh_n)^{-1}\}. \end{aligned}$$

All proofs are given in Section 2.8.

Theorem 2.1 *Use the same notation in Proposition 2.1 and assume all conditions and assumptions in Proposition 2.1 are satisfied. To simplify notation, let* $m_3 = 4m_1m_2 + m_1 + m_2$. *Furthermore, assume that* $K_1 \sim C_1 n^{\tau_1}, K_2 \sim C_2 n^{\tau_2}$ *with* $\tau_1 > (m_1+1)m_2/m_3$, $\tau_2 > m_1(m_2+1)/m_3$, $h_{n,1} \sim h_1 n^{-m_2/m_3}, h_{n,2} \sim h_2 n^{-m_1/m3}$ *for positive constants* C_1, C_2 *and* h_1, h_2. *Then, for any* $(x,z) \in (0,1) \times (0,1)$, *we have that*

$$n^{(2m_1m_2)/m_3}\left\{\hat{\mu}(x,z) - \mu(x,z)\right\} \Rightarrow N\left\{\tilde{\mu}(x,z), V(x,z)\right\} \tag{2.15}$$

in distribution as $n_1 \to \infty, n_2 \to \infty$, *where*

$$\tilde{\mu}(x,z) = (-1)^{m_1+1} h_1^{2m_1} \frac{\partial^{2m_1}}{\partial x^{2m_1}} \mu(x,z) + (-1)^{m_2+1} h_2^{2m_2} \frac{\partial^{2m_2}}{\partial z^{2m_2}} \mu(x,z), \tag{2.16}$$

$$V(x,z) = \sigma^2(x,z) \int H_{m_1}^2(u)\mathrm{d}u \int H_{m_2}^2(v)\mathrm{d}v. \tag{2.17}$$

Remark 2.1 *The case* $m_1 = m_2 = m$ *is important. The convergence rate of the estimator becomes* $n^{-m/(2m+1)}$. *Stone (1980) obtained the optimal rates of convergence for nonparametric estimators. For a bivariate smooth function* $\mu(x,z)$ *with*

continuous $2m$*th derivatives, the corresponding optimal rate of convergence for estimating* $\mu(x, z)$ *at any inner point of the unit square is* $n^{-m/(2m+1)}$*. Hence when* $m_1 = m_2 = m$*, the sandwich smoother achieves the optimal rate of convergence. Note that the bivariate kernel estimator with the product kernel* $H_m(x)H_m(z)$ *also has a convergence rate of* $n^{-m/(2m+1)}$*.*

Remark 2.2 *For the univariate case, the convergence rate of P-splines with an* m*th order difference penalty is* $n^{-2m/(4m+1)}$ *(see Wang et al., 2011). So the rate of convergence for the bivariate case is slower which shows the effect of "curse of dimensionality".*

Remark 2.3 *Theorem 2.1 shows that, provided it is fast enough, the divergence rate of the number of knots does not affect the asymptotic distribution. For practical usage, we recommend* $K_1 = \min\{n_1/2, 35\}$ *and* $K_2 = \min\{n_2/2, 35\}$*, so that every bin has at least 4 data points. Note that for univariate P-splines, a number of* $\min\{n/4, 35\}$ *knots was recommended by Ruppert (2002).*

2.4 Irregularly Spaced Data

Suppose the design points are random and we use the model $y_i = \mu(x_i, z_i) + \epsilon_i, i = 1, \ldots, n$, that is y_i, x_i, and z_i now have only a single index rather than i, j as before. Assume the design points $\{(x_1, z_1), \ldots, (x_n, z_n)\}$ are independent and sampled from a distribution $F(x, z)$ in $[0, 1]^2$. Then the sandwich smoother can not be directly applied to irregularly spaced data. A solution to this problem is to bin the data first. We partition $[0, 1]^2$ into an $I_1 \times I_2$ grid of equal-size rectangular bins, and let $\tilde{y}_{\kappa,\ell}$ be the mean of all y_i such that (x_i, z_i) is in the (κ, ℓ)th bin. If there are no data

in the (κ, ℓ)th bin, $\tilde{y}_{\kappa,\ell}$ is defined arbitrarily, e.g., by a nearest neighbor estimator (see below). Assuming $\tilde{y}_{\kappa,\ell}$ is a data point at $(\tilde{x}_\kappa, \tilde{z}_\ell)$, the center of the (κ, ℓ)th bin, we apply the sandwich smoother to the grid data $\tilde{\mathbf{Y}} = (\tilde{y}_{\kappa,\ell})_{1\le\kappa\le I_1, 1\le\ell\le I_2}$ to get

$$\hat{\boldsymbol{\theta}}^* = \left(\boldsymbol{\Lambda}_2^{-1} \otimes \boldsymbol{\Lambda}_1^{-1}\right) (\mathbf{B}_2 \otimes \mathbf{B}_1)^T \tilde{\mathbf{y}},$$

where $\tilde{\mathbf{y}} = \mathrm{vec}(\tilde{\mathbf{Y}})$. Then our penalized estimate is defined as

$$\hat{\mu}(x, z) = \sum_{\kappa=1}^{c_1} \sum_{\ell=1}^{c_2} \hat{\theta}^*_{k,\ell} B^1_\kappa(x) B^2_\ell(z).$$

2.4.1 Practical Implementation

For the above estimation procedure to work with the fast algorithm in Section 2.2.2, we need to handle the problem when there are no data in some bins due to sampling variation. If there are no data in the (κ, ℓ)th bin, one solution is to define $\tilde{y}_{\kappa,\ell}$ to be the mean of values in the neighboring bins. Doing this has no effect on asymptotics, since bins will eventually have data. For small samples, filling in empty cells this way allows the sandwich smoother to be calculated, but one might flag the estimates in the vicinity of empty bins as non-reliable.

Another solution is to use an algorithm which iterates between the data and the smoothing parameters as follows. Initially, we let $\tilde{y}_{\kappa,\ell} = 0$ if the (κ, ℓ)th bin has no data point. Another possibility is to let $\tilde{y}_{\kappa,\ell}$ be, for some $M > 0$, the average of the M values of y with (x, z) coordinates located closest to the center of the (κ, ℓ)th bin. To determine the smoothing parameters (λ_1, λ_2) that minimize GCV, we only calculate the sums of squared errors for the bins with data and ignore the bins with no data. This way we could get a pair of smoothing parameters. Then for the bins with no data, we replace the $\tilde{y}_{\kappa,\ell}$'s by the estimated value with this pair of smoothing parameters. Now with the updated data, we could obtain another

pair of smoothing parameters. We repeat the above procedure until reaching some convergence.

2.4.2 Asymptotic Theory

As before, we divide the unit interval into an $I_1 \times I_2$ grid and let $I = I_1 I_2$ be the number of bins.

Theorem 2.2 *Assume the following conditions are satisfied.*

1. *There exists a constant* $\delta > 0$ *such that* $\sup_i \mathrm{E}\left(|y_i|^{2+\delta}\right) < \infty$.
2. *The regression function* $\mu(x, z)$ *has continuous* $2m$*th order derivatives where* $m = \max(m_1, m_2)$.
3. *The design points* $\{(x_i, z_i)\}_{i=1}^n$ *are independent and sampled from a distribution* $F(x, z)$ *with a density function* $f(x, z)$ *and assume* $f(x, z)$ *is positive over* $[0, 1]^2$ *and has continuous first derivatives.*
4. *Conditional on* $\{(x_i, z_i)\}_{i=1}^n$, *the random errors* $\epsilon_i, 1 \leq i \leq n$, *are independent with mean 0 and conditional variance* $\sigma^2(x_i, z_i)$.
5. *The variance function* $\sigma^2(x, z)$ *is twice continuously differentiable.*
6. $I \sim c_I n^\tau$ *and* $I_1 \sim c_0 I_2$ *for some constants* c_I, c_0 *and* $\tau > (4m_1 m_2)/(4m_1 m_2 + m_1 + m_2)$.

Fix $(x, z) \in (0, 1)^2$. *Then with the same notation and assumptions as in Theorem 2.1, we have that*

$$n^{(2m_1 m_2)/m_3} \{\hat{\mu}(x, z) - \mu(x, z))\} \Rightarrow N\{\tilde{\mu}(x, z), V(x, z)/f(x, z)\}$$

in distribution as $n \to \infty$ *where* $\tilde{\mu}(x, z)$ *is defined in (2.16) and* $V(x, z)$ *is defined in (2.17).*

Remark 2.4 *We assume random design points in Theorem 2.2. For the fixed design points, the result in Theorem 2.2 still holds if we replace condition 3 with the following:* $\sup_{\kappa,\ell} |n_{\kappa,\ell}/(nI^{-1}) - f(\tilde{x}_\kappa, \tilde{z}_\ell)| = o(1)$ *where* $n_{\kappa,\ell}$ *is the number of data points in the* (κ, ℓ)*th bin and* $f(x, z)$ *is a continuous and positive function.*

2.5 Simulation Study

This section compares the proposed sandwich smoother, Eilers and Marx's P-splines implemented with a GLAM algorithm (E-M/GLAM) and Wood's thin-plate regression splines (TPRS) in terms of mean integrated square errors (MISEs) and computation speed. Section 2.5.1 shows that MISEs of the sandwich smoother and E-M/GLAM are roughly comparable and smaller than those of TPRS, while Section 2.5.2 illustrates the computational advantage of the sandwich smoother over the other smoothers.

There is no public code or software available for implementing E-M/GLAM, the code is self-written. We attach an document in Appendix C showing how the code is written.

2.5.1 Regression Function Estimation

Two test functions were used in the simulation study: $f_1(x,z) = \sin\{2\pi(x-.5)^3\}\cos(4\pi z)$ and

$$f_2(x,z) = \frac{0.75}{\pi\sigma_x\sigma_z}\exp\left\{-(x-0.2)^2/\sigma_x^2 - (z-0.3)^2/\sigma_z^2\right\} + \frac{0.45}{\pi\sigma_x\sigma_z}\exp\left\{-(x-0.7)^2/\sigma_x^2 - (z-0.8)^2/\sigma_z^2\right\},$$

where $\sigma_x = 0.3, \sigma_z = 0.4$. Note that f_2 was used in Wood (2003). The two true surfaces are shown in Figure 2.1.

Performances of the three smoothers were assessed at two sample sizes. In the smaller sample study, each test function was sampled on the 20×30 regular grid in the unit square, and random errors were iid $N(0, \sigma^2)$ with σ equal to 0.1 and 0.5. In each case, 100 replicate data sets were generated and, for each replicate data, the test function was fitted by the three estimators and the integrated squared error (ISE) was calculated. For the spline basis and knots settings, based on the recommendation in Remark 2.3, 10 and 15 equidistant knots were used for the x- and z-axis for the two P-spline estimators. Thus, a total of 150 knots were used to construct the B-spline basis. Cubic B-splines were used with a second order difference penalty. For the thin plate regression estimator (TPRS), we implemented the TPRS using the function "bam" in a R package "mgcv" developed by Simon Wood. In this study, TPRS with a rank of 150 (i.e., the basis dimension is 150) were used. For all three estimators, the smoothing parameters were chosen by generalized cross validation (GCV). The performances of the three estimators were evaluated by the MISEs (see Table 2.1) and also boxplots of the ISEs (see Figure 2.2).

From Table 2.1 we can see that the sandwich smoother did better than E-M for

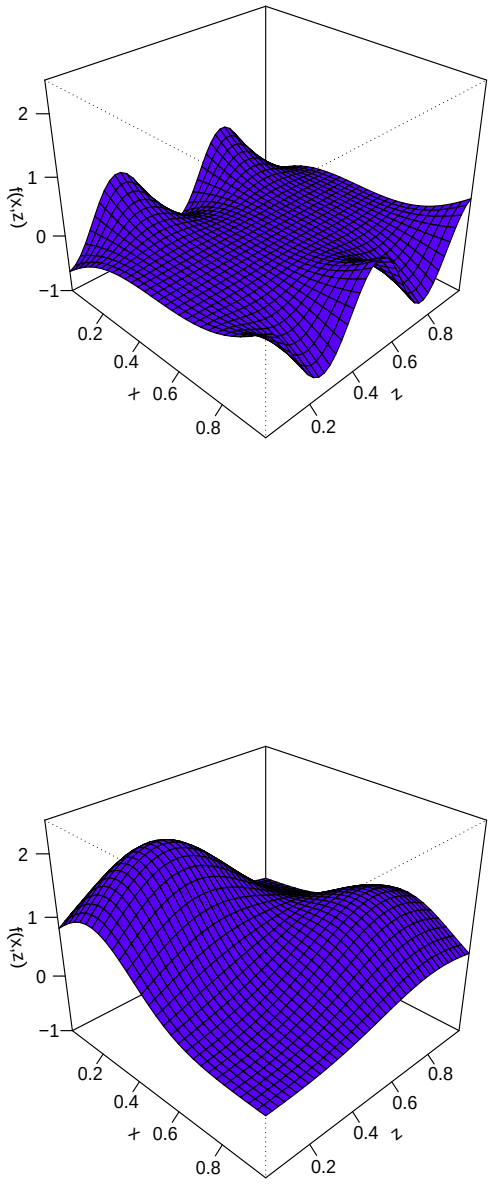


Figure 2.1: Surfaces of f_1 and f_2. The left surface is for f_1 and the right one is for f_2.

Table 2.1: MISEs of three estimators for small samples

	σ	Sandwich smoother	E-M/GLAM	TPRS
f_1	0.1	8.24×10^{-4}	9.53×10^{-4}	1.52×10^{-3}
	0.5	1.12×10^{-2}	1.22×10^{-2}	1.64×10^{-2}
f_2	0.1	6.60×10^{-4}	6.13×10^{-4}	6.94×10^{-4}
	0.5	9.88×10^{-3}	9.23×10^{-3}	8.60×10^{-3}

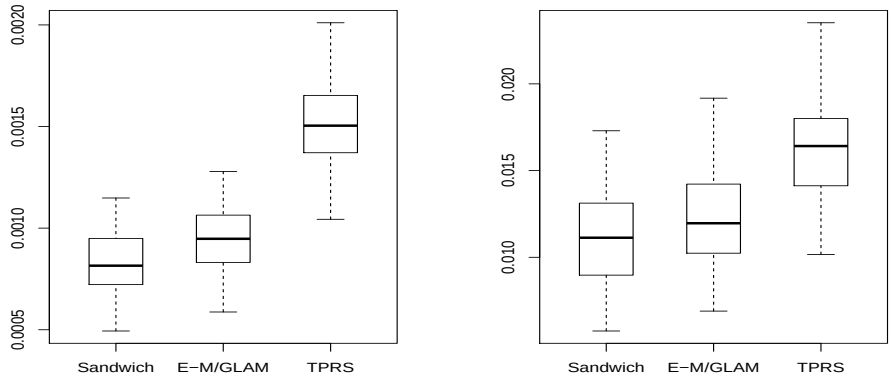
estimating f_1 while E-M was better for estimating f_2. The boxplots in Figure 2.2 show that the two P-spline methods are essentially comparable. Compared to the two P-spline methods, TPRS gave larger MISEs except for one case. One explanation for the relative inferior performance of TPRS for estimating f_1 is that TPRS is isotropic, which might be not appropriate for f_1 as f_1 is quite smooth in x and varies rapidly in z (see Figure 2.1).

A larger sample simulation study with $n_1 = 60$ and $n_2 = 80$ was also done. For the two P-spline estimators, the numbers of knots were $K_1 = 30$ and $K_2 = 35$. The rank of the TPRS was 1050, which was the total number of knots used in the two P-spline estimators. All the other settings were the same as in the smaller sample study. The resulting MISEs and boxplots gave the same conclusions as in the smaller sample study; see Table 2.2 and Figure 2.3.

2.5.2 Computation Speed

The computation speed of the three spline smoothers for smoothing f_2 with varying numbers of data points was assessed. For simplicity, we let $n_1 = n_2$ and considered the case $\sigma = 0.1$. We selected the number of knots for the two P-

(a) $f_1, \sigma = 0.1$ (b) $f_1, \sigma = 0.5$



(c) $f_2, \sigma = 0.1$ (d) $f_2, \sigma = 0.5$

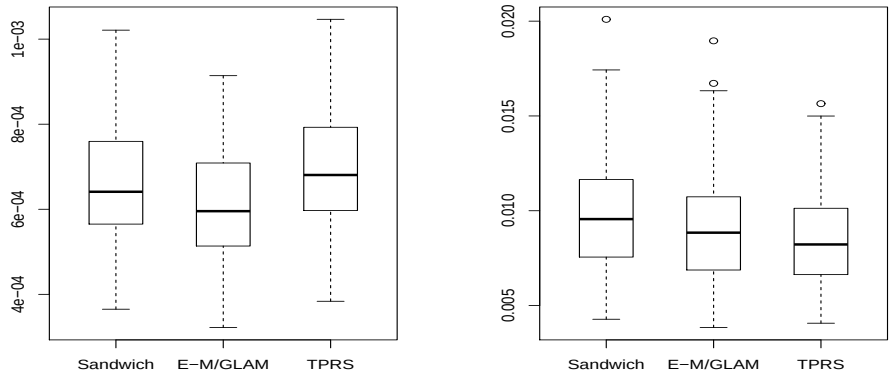


Figure 2.2: Boxplots of the ISEs of three estimators for small samples

spline smoothers following the recommendation in Remark 2.3. We fixed the rank of TPRS to the total number of knots used in the P-spline smoothers. For the two P-spline smoothers, the optimal smoothing parameters were searched over a 20×20 log scale grid in $[-5, 4]^2$. A finer grid with 40^2 grid points was also used. The computation was done on 2.83GHz computers running Windows with 3GB of RAM. Table 3.1 summarizes the results and shows that the sandwich smoother is

Table 2.2: MISEs of three estimators for larger samples

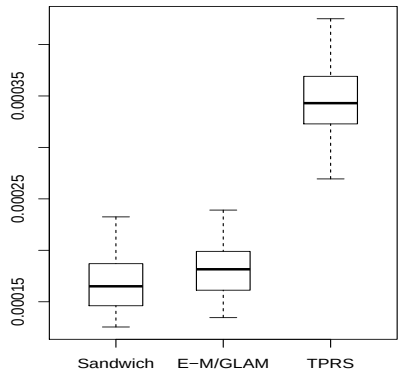
	σ	Sandwich smoother	E-M/GLAM	TPRS
f_1	0.1	1.68×10^{-4}	1.81×10^{-4}	3.46×10^{-4}
	0.5	2.16×10^{-3}	2.40×10^{-3}	3.66×10^{-3}
f_2	0.1	1.30×10^{-4}	1.21×10^{-4}	1.39×10^{-4}
	0.5	1.82×10^{-3}	1.71×10^{-3}	1.74×10^{-3}

the fastest method. Note that the values in parenthesis are the computation time using the finer grid.

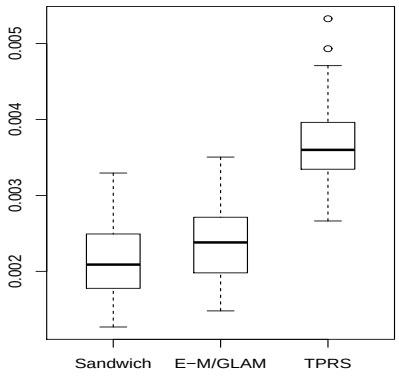
To further illustrate its computational capacity, the sandwich smoother was applied to large data with sizes of 300^2 and 500^2. For cubic B-splines coupled with second-order difference penalty, Theorem 2.1 suggested choosing $K_1 > n^{3/10}, K_2 > n^{3/10}$. So we let $K_1 = K_2$ with $K_1 K_2$ close to $n^{3/5+0.1}$ in the simulations. We also evaluated the speed of E-M/GLAM. To save time, the E-M/GLAM was run for only 25 pairs of smoothing parameters and the computation time was multiplied by 16 (64) so as to be comparable to that of the sandwich smoother. The results in Table 3.1 show that the sandwich smoother could process large data quite fast on a personal computer while the E-M/GLAM is much slower. The TPRS was not applied to these large data as it would require more memory space than the computer could provide.

To summarize, the simulation study here and also the fast algorithm in Section 2.2.2 show the advantage of the the sandwich smoother over the two other estimators. So when computation time is of concern, the sandwich smoother might be preferred.

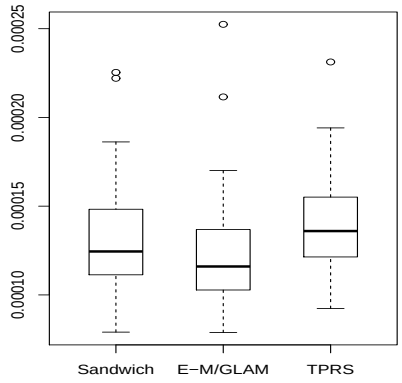
(a) $f_1, \sigma = 0.1$



(b) $f_1, \sigma = 0.5$



(c) $f_2, \sigma = 0.1$



(d) $f_2, \sigma = 0.5$

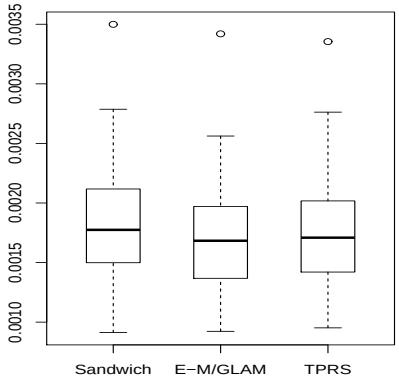


Figure 2.3: Boxplots of the ISEs of three estimators for larger samples

2.6 Application: Covariance Function Estimation

As functional data analysis (FDA) has become a major research area, estimation of covariance functions has become an important application of bivariate smoothing. Therefore, fast calculation of bivariate smooths is essential in FDA, especially when the bootstrap is used for inference. Local polynomial smoothing is a popular

Table 2.3: Computation time (in seconds) of three estimators averaged over 100 data sets on 2.83GHz computers running Windows with 3GB of RAM. The values in parenthesis are for the finer grid. For $n = 20^2, 40^2$ and 80^2, the number of knots for each axis is chosen by the recommendation in Remark 2.3. For $n = 300^2$ and 500^2, the total number of knots for the sandwich smoother is approximately $n^{3/5+0.1}$ as suggested by Theorem 2.1.

n	K_1K_2	Sandwich smoother	E-M/GLAM	TPRS
20^2	10^2	0.06(0.24)	4.09(19.74)	0.53
40^2	20^2	0.08(0.30)	99.88(413.98)	19.50
80^2	35^2	0.13(0.45)	1847.90(6906.76)	886.37
300^2	42^2	0.18(0.58)	5673(22696)	–
500^2	57^2	0.32(0.89)	38047(152188)	–

method in estimating covariance functions (see e.g., Yao *et al.*, 2005; Yao and Lee, 2006) while other smoothing methods such as kernel (Staniswalis and Lee, 1998) and penalized splines (Di *et al.*, 2009) have also been used. In this section, first through a simulation study we compare the performance of the sandwich smoother and local polynomial smoothers for estimating a covariance function when the data are observed or measured at a fixed grid, then we give a real data example.

2.6.1 Simulation Study

Let $\{X(t) : t \in [0, 1]\}$ be a stochastic process with a continuous covariance function $K(s, t) = \text{cov}\{X(s), X(t)\}$. For simplicity, we assume $\text{E}X(t) = 0, t \in [0, 1]$. Sup-

pose $\{X_i(t), i = 1, \ldots, n\}$ is a collection of independent realizations of the above stochastic process and we observe the random functions X_i at discrete design points with measurement errors,

$$Y_{ij} = X_i(t_j) + \epsilon_{ij}, 1 \leq j \leq m, 1 \leq i \leq n,$$

where m is the number of measurements per curve, n is the total number of curves, and the ϵ_{ij} are i.i.d. measurement errors with mean zero and finite variance and they are independent from the random functions X_i. Let $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{im})^T$. Then an estimate of the covariance function can be obtained through smoothing the sample covariance matrix $n^{-1} \sum_{i=1}^{n} \mathbf{Y}_i \mathbf{Y}_i^T$ by a bivariate smoother. For the sandwich smoother, we use two identical univariate smoother matrices and there is only one smoothing parameter to select. We use the commonly used local linear smoother (Yao *et al.*, 2005, Hall *et al.*, 2006) for comparison and the bandwidth is selected by the leave-one-curve-out cross validation. We use our own code which gives the same estimator used by Yao *et al.* (2005) but we do not use their code because it is in Matlab.

We let $K(s,t) = \sum_{k=1}^{4} \lambda_k \psi_k(s) \psi_k(t)$ where the eigenvalues $\lambda_k = 0.5^{k-1}, k = 1, 2, 3, 4$, and $\{\psi_1, \ldots, \psi_4\}$ are the eigenfunctions from either of the following

$$\text{Case 1:} \quad \left\{\sqrt{2}\sin(2\pi t), \sqrt{2}\cos(2\pi t), \sqrt{2}\sin(4\pi t), \sqrt{2}\cos(4\pi t)\right\},$$

$$\text{Case 2:} \quad \left\{1, \sqrt{3}(2t-1), \sqrt{5}(6t^2 - 6t + 1), \sqrt{7}(20t^3 - 30t^2 + 12t - 1)\right\}.$$

The above two sets of eigenfunctions were used in Di *et al.* (2009), Greven *et al.* (2010), and Zipunnikov *et al.* (2011). We let $\sigma = 0.5$. We simulate 100 datasets and evaluate the two bivariate smoothers in terms of mean ISEs (MISEs). The results are given in Table 2.4. Table 2.4 shows that the two smoother have quite close MISEs for all cases. The estimated eigenfunctions by the two smoothers with $(n, m) = (25, 20)$ are shown in Figures 2.4 and 2.5. The figure shows that

Table 2.4: MISEs of the sandwich smoother and the local linear smoother for estimating a covariance function. The number in parenthesis is the standard deviation of ISE's.

(n, m)	Case	Sandwich smoother	Local linear smoother
$(25, 20)$	1	.053(.035)	.050(.026)
	2	.199(.139)	.204(.144)
$(100, 40)$	1	.014(.008)	.013(.008)
	2	.050(.034)	.050(.036)

both smoothers well estimate the eigenfunctions. We find the same results for $(n, m) = (100, 40)$ and hence do not show the figure in the paper.

We also compare the computation time of the three smoothers using Case 1 for various m; see Table 2.5. The sandwich smoother is seen to be much faster to compute than the local linear smoother for covariance function estimation.

To summarize, the above simulation study suggests the sandwich smoother is comparable to the local linear smoother in terms of MISEs for covariance function estimation when the functional data are measured at a fixed grid. The sandwich smoother is also considerably faster to compute than the local linear smoother.

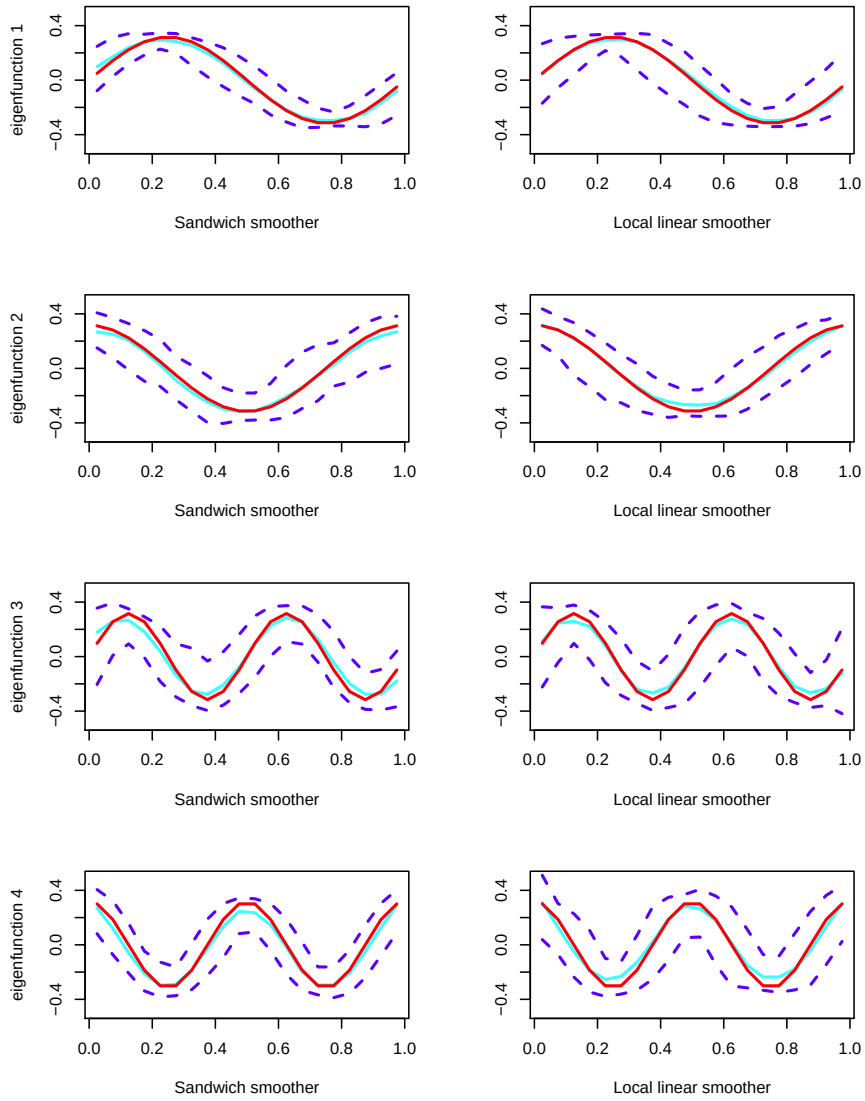


Figure 2.4: True and estimated eigenfunctions replicated 100 times with $(n, m) = (25, 20)$ for case 1. The variance of noises is 0.25. Each box shows the pointwise median estimated eigenfunction (cyan solid lines), the true eigenfunction (solid red lines), the 5th and 95th pointwise percentile curves (dashed blue lines). The left column is for the sandwich smoother and the right one is for local linear smoother.

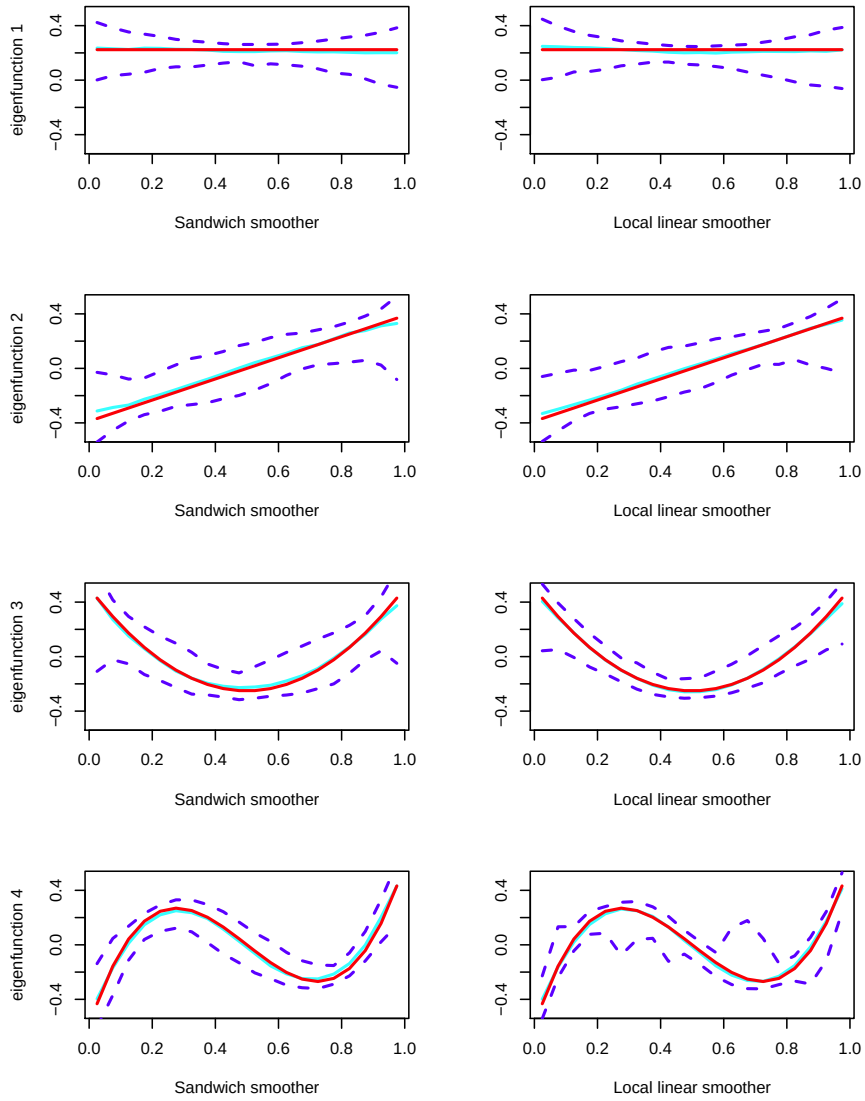


Figure 2.5: True and estimated eigenfunctions replicated 100 times with $(n, m) = (25, 20)$ for case 2. The variance of noises is 0.25. Each box shows the pointwise median estimated eigenfunction (cyan solid lines), the true eigenfunction (solid red lines), the 5th and 95th pointwise percentile curves (dashed blue lines). The left column is for the sandwich smoother and the right one is for local linear smoother.

Table 2.5: Computation time (in seconds) of the sandwich smoother and the local linear smoother averaged over 100 data sets on 2.83GHz computers running Windows with 3GB of RAM. The number of curves is fixed at 100. The bandwidth for the local linear smoother is fixed in the computations.

m	Sandwich smoother	Local linear smoother
40	0.02	2.98
80	0.03	50.04
160	0.05	961.42
320	0.16	13854.40

2.6.2 Example: Estimating a Covariance Function in Diffusion Tensor Imaging Data

The Diffusion Tensor Imaging (DTI) data comes from magnetic resonance imaging (MRI) scans of the brain and spinal cord of multiple sclerosis (MS) patients and controls. It has been shown in some animal studies that the variability of one DTI index, fractional anisotropy (FA), increases in MS plaques (Tievsky *et al.*, 1999; Werring *et al.*, 1999). Here we study FA to facilitate detecting or monitoring pathologic changes in MS lesions. The DTI data is from the R package "refund" by Crainiceanu and Reiss (2012) and has been analyzed by Goldsmith *et al.* (2011) and Goldsmith *et al.* (2012).

We study FA as a function of location along the corpus callosum (CCA), a tract that connects the left and right hemispheres of the brain. The tract profiles are

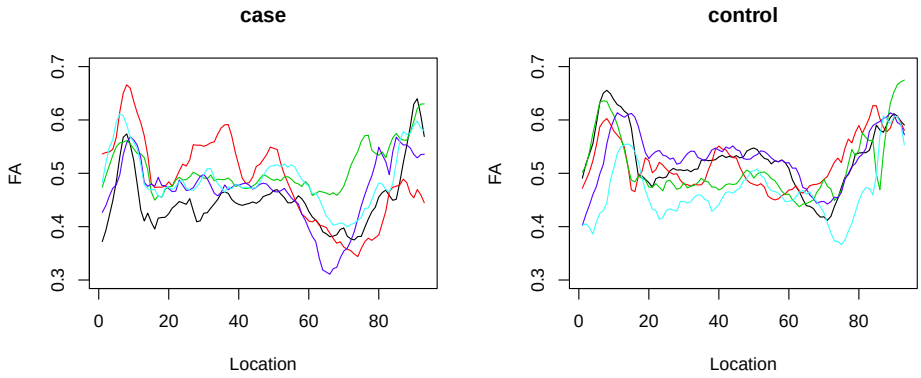


Figure 2.6: Five random selected curves in the case and control groups. The data source is the R package "refund" by Crainiceanu and Reiss (2012)

spatially normalized and each subject has 93 equidistant sample points from the splenium to the genu. We use FA data collected from 100 patients and 42 controls during their first visit. Examples of FA profiles are shown in Figure 2.6.

Let $X_i(t)$ be a smooth random curve for the ith FA profile with mean $\mu(t)$ and covariance function $K(s,t) = \text{cov}\{X_i(s), X_i(t)\}$. By Mercer's theorem, we can express $K(s,t)$ as $\sum_{\kappa=1}^{\infty} \lambda_\kappa \psi_\kappa(s)\psi_\kappa(t)$, where λ_κ are the ordered nonnegative eigenvalues and ψ_κ's are the corresponding orthogonal eigenfunctions with unit L_2 norm. Then the Karhunen-Loève expansion implies that $X_i(t) = \mu(t) + \sum_{\kappa=1}^{\infty} \xi_{i\kappa}\psi_\kappa(t)$, where $\xi_{i\kappa} = \int\{X_i(t) - \mu(t)\}\psi_\kappa(t)dt$ are uncorrelated random variables with mean 0 and variance λ_κ. Let Y_{ij} be the noisy observation of X_i at location t_j with measurement error e_{ij}. The functional data model is

$$Y_{ij} = X_i(t_j) + e_{ij} = \mu(t_j) + \sum_{\kappa=1}^{\infty} \xi_{i\kappa}\psi_\kappa(t_j) + e_{ij},\ 1 \le i \le 142, 1 \le j \le 93.$$

Here e_{ij} are assumed independent with mean 0 and variance σ^2.

We do not distinguish cases from controls in the notation because, as a working assumption, we use a common covariance function for cases and controls, so that

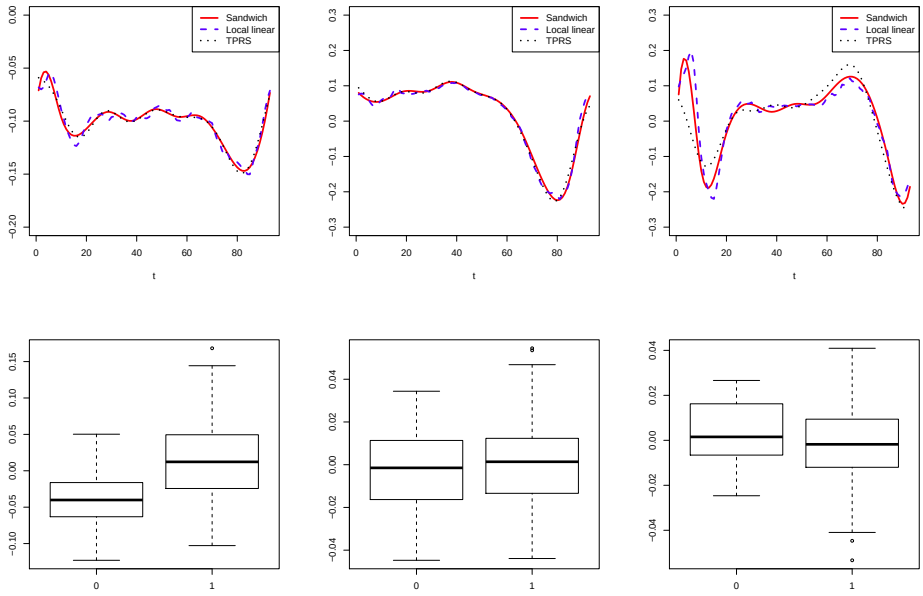


Figure 2.7: The top row provides plots of the estimated eigenfunctions by the sandwich smoother, TPRS, and the local linear smoother. The bottom row provides boxplots of principal scores obtained by the sandwich smoother. Group 0 refers to control and Group 1 refers to cases. The first, second, and third eigenfunctions and their principal scores are in the left, middle, and right columns, respectively.

they can be compared using their scores on the common eigenfunctions. All that is really needed here is that the two covariance functions have the same eigenvectors. This assumption is plausible, at least as an approximation.

As in the estimation procedure in Yao and Lee (2006), we apply the sandwich smoother, the local linear smoother, and thin plate regression splines to estimate the covariance function and to extract the eigenfunctions. The first five eigenfunctions eigenfunctions explain more than 90% of the variations. Here we focus on the first three eigenfunctions. For the sandwich smoother the first three eigenfunctions account for $68\%, 8\%$ and 7% of the variations, respectively. The percentages are $71\%, 8\%, 7\%$ for TPRS and $66\%, 9\%$ and 6% for the local linear smoother. Esti-

mates of the first three eigenfunctions are shown on the top of Figure 2.7. From Figure 2.7, we see that all three smoother give essentially the same eigenfunctions. Since only small differences exist between the principal scores from all the approaches, we only discuss the principal scores obtained by the sandwich smoother. The bottom of Figure 2.7 contains boxplots of the principal scores associated with the corresponding eigenfunctions for the case and control groups. We can see that tract profiles from the case group tend to have higher first principal scores. The above observation can be verified by the Mann-Whitney-Wilcoxon test, which gives a p-value of 8.8×10^{-10}. So when MS is present, the first principal scores tend to be larger. The second eigenfunction characterizes the mean contrast of tract profiles between a valley around 80 and the other places. The third eigenfunction characterizes the mean contrast of tract profiles between a valley near 15 and a bump around 70. So this study shows that the case group exhibits more variations in the principal scores, i.e., FA variability tends to increase when MS is present.

2.7 Multivariate P-splines

We extend the sandwich smoother to array data of dimensions greater than two. Suppose we have a nonparametric regression model with $d \geq 3$ covariates

$$y_{i_1,\ldots,i_d} = \mu(x_{i_1}, \ldots, x_{i_d}) + \epsilon_{i_1,\ldots,i_d}, \quad 1 \leq i_k \leq n_k, 1 \leq k \leq d,$$

so the data are collected on a d-dimensional grid. For simplicity, assume the covariates are in $[0,1]^d$. As in the bivariate case, we model the d-variate function $\mu(x_1, \ldots, x_d)$ by tensor product B-splines of d variables $\sum_{\kappa_1,\kappa_2,\ldots,\kappa_d} \theta_{\kappa_1,\kappa_2,\ldots,\kappa_d} B^1_{\kappa_1}(x_1) B^1_{\kappa_2}(x_2) \cdots B^d_{\kappa_d}(x_d)$, where $B^1_{\kappa_1}, B^2_{\kappa_2}, \ldots, B^d_{\kappa_d}$ are B-spline basis functions. We smooth along all covariates simultaneously so that

the fitted values and the data satisfy

$$\hat{\mathbf{y}} = (\mathbf{S}_d \otimes \mathbf{S}_{d-1} \otimes \cdots \otimes \mathbf{S}_1)\,\mathbf{y}, \tag{2.18}$$

where $\mathbf{S}_i$ is the smoother matrix for the ith covariate using P-splines as in (2.3), $\mathbf{y}$ is the data vector organized first by x_1, then by x_2, and so on, and $\hat{\mathbf{y}}$ is organized the same way as $\mathbf{y}$. Similar to equation (2.7), the estimate of coefficients $\hat{\boldsymbol{\theta}}$ satisfies

$$(\boldsymbol{\Lambda}_d \otimes \boldsymbol{\Lambda}_{d-1} \otimes \cdots \otimes \boldsymbol{\Lambda}_1)\,\hat{\boldsymbol{\theta}} = (\mathbf{B}_d \otimes \mathbf{B}_{d-1} \otimes \cdots \otimes \mathbf{B}_1)^T \mathbf{y},$$

and the penalized estimate is

$$\hat{\mu}(x_1, x_2, \ldots, x_d) = \sum_{\kappa_1,\kappa_2,\ldots,\kappa_d} \hat{\theta}_{\kappa_1,\kappa_2,\ldots,\kappa_d} B^1_{\kappa_1}(x_1) B^1_{\kappa_2}(x_2) \cdots B^d_{\kappa_d}(x_d).$$

2.7.1 Fast Algorithm

Two computational issues occur for smoothing data on an multi-dimensional grid. The first issue is that unless the sizes of $\mathbf{S}_i$'s are all small, the storage and computation of $\mathbf{S}_d \otimes \mathbf{S}_{d-1} \otimes \cdots \otimes \mathbf{S}_1$ will be challenging. The second issue is selection of smoothing parameters. Because several smoothing parameters are involved, finding the smoothing parameters that minimize some model selection criteria such as GCV can be difficult.

The generalized linear array model (GLAM) by Currie *et al.* (2006) solved the first issue by making use of the array structures of the model matrix as well as the data. The smoother matrix $\mathbf{S}_d \otimes \mathbf{S}_{d-1} \otimes \cdots \otimes \mathbf{S}_1$ in multivariate smoothing has a tensor product structure, hence $\hat{\mathbf{y}}$ in (2.18) can be computed efficiently by a sequence of nested operations on $\mathbf{y}$ as in the GLAM. For instance, consider $d = 3$. Let $\mathbf{Y}$ be the $n_1 \times n_2 \times \cdots \times n_d$ dimensional data array, then $\hat{\mathbf{y}}$ can be computed efficiently with one line of R code:

```
# The function "RH" is rotated H-transform of an array by a matrix
# see Currie et al. (2006)
yhat = as.vector(RH(S3,RH(S2,RH(S1,Y))))
```

For the second issue, because of the tensor product structure of the smoother matrix, the fast algorithm derived in Section 2.2.2 can be easily generalized for the multivariate case. As an illustration, we show how to compute the trace of the smoother matrix. We first compute the singular value decompositions for all $\mathbf{S}_i$ so that (2.13) holds for all $i = 1, \ldots, d$, then we compute the trace of the smoother matrix by

$$tr\left(\mathbf{S}_d \otimes \mathbf{S}_{d-1} \otimes \cdots \otimes \mathbf{S}_1\right) = \prod_{i=1}^{d} tr(\mathbf{S}_i)$$

using the identity in (2.12) repeatedly. Note that $tr(\mathbf{S}_i)$ has a similar expression as in (2.13) for all i.

2.7.2 Comparison with GLAM for Smoothing

GLAMs handle array structures efficiently and can be used with data from the exponential family. the sandwich smoother method in this paper is less widely applicable because it does not allow for arbitrary weights. Thus, the sandwich smoother can be used whenever least squares, rather than of generalized least squares, is appropriate. One example is covariance function estimation in functional data analysis; see Section 2.6.2. So in the context of smoothing, the sandwich smoother is less general than GLAM. However, when applicable, the sandwich smoother is preferred whenever the computational burden is a concern.

For instance, for smoothing some image data of size $128 \times 128 \times 24$, the sandwich smoother takes about 20 seconds on a 2.83GHz computer running windows with

3GB of RAM. We have not found the computation time of other algorithms, but we can give a crude lower bound. We see in Table 3.1 that E-M/GLAM takes 1848 seconds (over 30 minutes) on an 80×80 grid where the smoothing parameters are searched over a 20×20 grid. Searching over a $20 \times 20 \times 20$ grid to select the smoothing parameters, the number of times of GCV computation is now 20 times more. Moreover, for each GCV computation, E-M/GLAM will need much more time for smoothing data of size $128 \times 128 \times 24$ which is much larger. Therefore, we expect that the computation time for smoothing a $128 \times 128 \times 24$ will be many hours for an algorithm that does not compute GCV as efficiently as the sandwich smoother does.

2.8 Proof of Theorems

Lemma 2.1 *Suppose* $\mathbf{A}, \mathbf{B}$ *and* $\mathbf{X}$ *are matrices of compatible dimensions. Then*

$$\text{vec}\,(\mathbf{AXB}) = \left(\mathbf{A} \otimes \mathbf{B}^T\right) \text{vec}(\mathbf{X}).$$

Proof of Lemma 2.1: see page 240 of Seber (2007).

Lemma 2.2 *Suppose* $\mathbf{A}, \mathbf{B}, \mathbf{C}$ *and* $\mathbf{D}$ *are matrices of compatible dimensions. Then*

$$(\mathbf{AB}) \otimes (\mathbf{CD}) = (\mathbf{A} \otimes \mathbf{C})(\mathbf{B} \otimes \mathbf{D}),$$

$$(\mathbf{A} \otimes \mathbf{C})^T = \mathbf{A}^T \otimes \mathbf{C}^T.$$

If $\mathbf{F}$ *and* $\mathbf{G}$ *are invertible square matrices, then*

$$(\mathbf{F} \otimes \mathbf{G})^{-1} = \mathbf{F}^{-1} \otimes \mathbf{G}^{-1}.$$

Proof of Lemma 2.2: see pages 235- 239 of Seber (2007).

Lemma 2.3 *Suppose* $\mathbf{A}$ *and* $\mathbf{B}$ *are square matrices. Then*

$$\mathrm{tr}\,(\mathbf{A}\otimes\mathbf{B}) = \mathrm{tr}(\mathbf{A})\cdot\mathrm{tr}(\mathbf{B}).$$

Proof of Lemma 2.3: see page 235 of Seber (2007).

Before proving Proposition 2.1, we need the following lemma:

Lemma 2.4 *Use the same notation in Proposition 2.1 and assume all conditions and assumptions in Proposition 2.1 are satisfied. For* $(x,z)\in(0,1)\times(0,1)$*, there exists a constant* $C>0$ *such that*

$$\hat{\mu}(x,z) = \sum_{i,j} y_{i,j}\left[\left\{\sum_{\kappa,r} B^1_\kappa(x)B^1_r(x_i)S_{\kappa,r,x}\right\}\left\{\sum_{\ell,s} B^2_\ell(z)B^2_s(z_j)S_{\ell,s,z}\right\} + \tilde{b}_{i,j}(x,z)\right],$$

where $\tilde{b}_{i,j}(x,z) = O\left[\exp\left\{-C\min(h_{n,1}^{-1},h_{n,2}^{-1})\right\}\right]$.

Proof of Lemma 2.4: By (2.8), $\hat{\mu}(x,z) = \sum\hat{\theta}_{\kappa,\ell}B^1_\kappa(x)B^2_\ell(z)$. We only need to consider $\hat{\theta}_{\kappa,\ell}$ for which $B^1_\kappa(x)$ and $B^2_\ell(z)$ are both non-zero. Hence assume κ and ℓ satisfy $\kappa\in(K_1x-p_1-1,K_1x+p_1+1)$, $\ell\in(K_2z-p_2-1,K_2z+p_2+1)$. Let $q_1=\max(p_1,m_1)$ and $q_2=\max(p_2,m_2)$. Denote by $\mathbf{\Lambda}_{1,j}$ the jth column of $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_{2,j}$ the jth column of $\mathbf{\Lambda}_2$. As shown in Section A.4 of Appendix A and Li and Ruppert (2008), there exist vectors $\mathbf{S}_{\kappa,x}$ and a constant $C_3>0$ so that for $q_1<j<c_1-q_1$, $\mathbf{S}^T_{\kappa,x}\mathbf{\Lambda}_{1,j}=\delta_{\kappa,j}$, and for $1\le j\le q_1$ or $c_1-q_1\le j\le c_1$, $\mathbf{S}^T_{\kappa,x}\mathbf{\Lambda}_{1,j} = O\left[\exp\left\{-C_3h_{n,1}^{-1}\min(x,1-x)\right\}\right]$. Here $\delta_{\kappa,j}=1$ if $j=\kappa$ and 0 otherwise. Similarly, there exist vectors $\mathbf{S}_{\ell,z}$ and a constant $C_4>0$ such that for $q_2<j<c_2-q_2$, $\mathbf{S}^T_{\ell,z}\mathbf{\Lambda}_{2,j}=\delta_{\ell,j}$, and for $1\le j\le q_2$ or $c_2-q_2\le j\le c_2$, $\mathbf{S}^T_{\ell,z}\mathbf{\Lambda}_{2,j} = O\left[\exp\left\{-C_4h_{n,2}^{-1}\min(z,1-z)\right\}\right]$. Let $\tilde{\theta}_{\kappa,\ell} = \left(\mathbf{S}_{\ell,z}\otimes\mathbf{S}_{\kappa,x}\right)^T\left(\mathbf{\Lambda}_2\otimes\mathbf{\Lambda}_1\right)\hat{\boldsymbol{\theta}}$ and $C=\min\left\{C_3\min(x,1-x),C_4\min(z,1-z)\right\}$, then

$$\tilde{\theta}_{\kappa,\ell}-\hat{\theta}_{\kappa,\ell} = \sum_{i,j}\tilde{b}_{i,j,\kappa,\ell}y_{i,j}, \tag{2.19}$$

where $\tilde{b}_{i,j,\kappa,\ell} = O\left[\exp\left\{-C\min(h_{n,1}^{-1}, h_{n,2}^{-1})\right\}\right]$. By equation (2.7),

$$\tilde{\theta}_{\kappa,\ell} = \left(\mathbf{S}_{\ell,z} \otimes \mathbf{S}_{\kappa,x}\right)^T \left(\mathbf{B}_2^T \otimes \mathbf{B}_1^T\right) \mathbf{y} = \left(\mathbf{S}_{\ell,z}^T \mathbf{B}_2^T \otimes \mathbf{S}_{\kappa,x}^T \mathbf{B}_1^T\right) \mathbf{y} = \mathbf{S}_{\kappa,x}^T \left(\mathbf{B}_1^T \mathbf{Y} \mathbf{B}_2\right) \mathbf{S}_{\ell,z}.$$

Letting $S_{\kappa,r,x}$ be the rth element of $\mathbf{S}_{\kappa,x}$ and similarly $S_{\ell,s,z}$ the sth element of $\mathbf{S}_{\ell,z}$, we express $\tilde{\theta}_{\kappa,\ell}$ as a double sum

$$\begin{aligned}\tilde{\theta}_{\kappa,\ell} &= \sum_{r,s} S_{\kappa,r,x} \left\{ \sum_{i,j} B_r^1(x_i) y_{i,j} B_s^2(z_j) \right\} S_{\ell,s,z} \\ &= \sum_{i,j} y_{i,j} \left\{ \sum_r B_r^1(x_i) S_{\kappa,r,x} \right\} \left\{ \sum_s B_s^2(z_j) S_{\ell,s,z} \right\}.\end{aligned}$$

With the above equation and (2.8), (2.19), we have

$$\begin{aligned}\hat{\mu}(x,z) &= \sum_{\kappa,\ell} \tilde{\theta}_{\kappa,\ell} B_\kappa^1(x) B_\ell^2(z) + \sum_{\kappa,\ell} (\hat{\theta}_{\kappa,\ell} - \tilde{\theta}_{\kappa,\ell}) B_\kappa^1(x) B_\ell^2(z) \\ &= \sum_{i,j} y_{i,j} \left[\left\{ \sum_{\kappa,r} B_\kappa^1(x) B_r^1(x_i) S_{\kappa,r,x} \right\} \left\{ \sum_{\ell,s} B_\ell^2(z) B_s^2(z_j) S_{\ell,s,z} \right\} + \tilde{b}_{i,j}(x,z) \right],\end{aligned}$$

where $\tilde{b}_{i,j}(x,z) = O\left[\exp\left\{-C\min(h_{n,1}^{-1}, h_{n,2}^{-1})\right\}\right]$.

Proof of Proposition 2.1: Let $\tilde{\lambda}_1 = \lambda_1 K_1 n_1^{-1} = (K_1 h_{n,1})^{2m_1}$ and $\tilde{\lambda}_2 = \lambda_2 K_2 n_2^{-1} = (K_2 h_{n,2})^{2m_2}$. By Proposition A.5 in Section A.5 of Appendix A, there exists some constants $0 < \phi_1, \phi_2 < \infty$ such that

$$\begin{aligned}&n_1 h_{n,1} \sum_{k,r} B_k^1(x) B_r^1(x_i) S_{k,r,x} \\ =&H_{m_1}\left(\frac{|x-x_i|}{h_{n,1}}\right) + \delta_{\{p_1>m_1\}} \left[O\left(\tilde{\lambda}_1^{-2+\frac{1}{2m_1}}\right) + \delta_{\{|x-x_i|<\phi_1/K_1\}} O\left(\tilde{\lambda}_1^{-\frac{p_1}{p_1-m_1}+\frac{1}{2m_1}}\right)\right] \\ &+ \exp\left(-\phi_2 \frac{|x-x_i|}{h_{n,1}}\right) \left[O\left(\tilde{\lambda}_1^{-\frac{1}{m_1}}\right) + \delta_{\{m_1=1\}} \delta_{\left\{|x-x_i| \le (p_1+1)\tilde{\lambda}_1^{-1/(2m_1)}\right\}} O\left(\tilde{\lambda}_1^{-\frac{1}{2m_1}}\right)\right]\end{aligned} \tag{2.20}$$

Here $\delta_{\{p_1>m_1\}} = 1$ if $p_1 > m_1$ and 0 otherwise; the other δ terms are similarly

defined. Similarly, there exist some constants $0 < \phi_3, \phi_4 < \infty$ such that

$$
\begin{aligned}
& n_2 h_{n,2} \sum_{\ell,s} B_\ell^2(z) B_s^2(z_j) S_{\ell,s,z} \\
=& H_{m_2}\left(\frac{|z-z_j|}{h_{n,2}}\right) + \delta_{\{p_2>m_2\}}\left[O\left(\tilde{\lambda}_2^{-2+\frac{1}{2m_2}}\right) + \delta_{\{|z-z_j|<\phi_3/K_2\}} O\left(\tilde{\lambda}_2^{-\frac{p_2}{p_2-m_2}+\frac{1}{2m_2}}\right)\right] \\
& + \exp\left(-\phi_4 \frac{|z-z_j|}{h_{n,2}}\right)\left[O\left(\tilde{\lambda}_2^{-\frac{1}{m_2}}\right) + \delta_{\{m_2=1\}}\delta_{\left\{|z-z_j|\le(p_2+1)\tilde{\lambda}_2^{-1/(2m_2)}\right\}} O\left(\tilde{\lambda}_2^{-\frac{1}{2m_2}}\right)\right].
\end{aligned}
\tag{2.21}
$$

Let

$$
\begin{aligned}
d_{i,1} &= \sum_{k,r} B_k^1(x) B_r^1(x_i) S_{k,r,x} - (n_1 h_{n,1})^{-1} H_{m_1}\left\{h_{n,1}^{-1}(x-x_i)\right\}, \\
d_{i,2} &= \sum_{\ell,s} B_\ell^2(z) B_s^2(z_j) S_{\ell,s,z} - (n_2 h_{n,2})^{-1} H_{m_2}\left\{h_{n,2}^{-1}(z-z_j)\right\}, \\
b_{i,j}(x,z) &= \frac{d_{i,2}}{n_1 h_{n,1}} H_{m_1}\left(\frac{|x-x_i|}{h_{n,1}}\right) + \frac{d_{i,1}}{n_2 h_{n,2}} H_{m_2}\left(\frac{|z-z_j|}{h_{n,2}}\right) + d_{i,1} d_{i,2} + \tilde{b}_{i,j}(x,z).
\end{aligned}
$$

It follows from Lemma 2.4 that $\hat{\mu}(x,z) - \mu^*(x,z) = \sum_{i,j} b_{i,j}(x,z) y_{i,j}$. Hence $\mathrm{E}\{\hat{\mu}(x,z) - \mu^*(x,z)\} = \sum_{i,j} b_{i,j}(x,z)\mu(x_i,z_j)$ and $\mathrm{var}\{\hat{\mu}(x,z) - \mu^*(x,z)\} = \sum_{i,j} b_{i,j}^2(x,z)\sigma^2(x_i,z_j)$.

To simplify notation, denote $\max\{(K_1 h_{n,1})^{-2}, (K_2 h_{n,2})^{-2}\}$ by ξ. We prove $\mathrm{E}\{\hat{\mu}(x,z) - \mu^*(x,z)\} = O(\xi)$ by showing that $\sum_{i,j} |b_{i,j}(x,z)\mu(x_i,z_j)|$ is $O(\xi)$. By Lemma 2.4, $\tilde{b}_{i,j}(x,z) = O\left[\exp\left\{-C\min(h_{n,1}^{-1}, h_{n,2}^{-1})\right\}\right]$. Since $h_{n,1} = O(n^{-\nu_1})$ and $h_{n,2} = O(n^{-\nu_2})$, $\tilde{b}_{i,j}(x,z) = n^{-1}o(\xi)$ and hence $\sum_{i,j} |\tilde{b}_{i,j}(x,z)\mu(x_i,z_j)| = o(\xi)$. For simplicity, we shall only show that

$$
\sum_{i,j}\left|\frac{1}{n_1 h_{n,1}} H_{m_1}\left(\frac{|x-x_i|}{h_{n,1}}\right) d_{i,2}\mu(x_i,z_j)\right| = O(\xi), \tag{2.22}
$$

and we use the case when $p_2 \leq m_2$ as an example. Because

$$\frac{1}{nh_n}\sum_{i,j}\left|H_{m_1}\left(\frac{|x-x_i|}{h_{n,1}}\right)\exp\left(-\phi_4\frac{|z-z_j|}{h_{n,2}}\right)\mu(x_i,z_j)\right| = O(1),$$

$$\frac{1}{nh_n}\sum_{i,j}\left|H_{m_1}\left(\frac{|x-x_i|}{h_{n,1}}\right)\exp\left(-\phi_4\frac{|z-z_j|}{h_{n,2}}\right)\delta_{\left\{|z-z_j|\leq(p_2+1)\tilde{\lambda}_2^{-1/(2m_2)}\right\}}\mu(x_i,z_j)\right|$$
$$= O\left\{\tilde{\lambda}_2^{-\frac{1}{2m_2}}\right\},$$

and $\tilde{\lambda}_2^{-1/m_2} = (K_2 h_{n,2})^{-2}$, equality (2.22) is proved. The case when $p_2 > m_2$ and the desired results involving $d_{i,1}$ can be similarly proved.

Next we show that $\mathrm{var}\{\hat{\mu}(x,z) - \mu^*(x,z)\} = o\{(nh_n)^{-1}\}$, i.e., $\sum_{i,j} b_{i,j}^2(x,z)\sigma^2(x_i,z_j) = o\{(nh_n)^{-1}\}$. Note that $b_{i,j}^2(x,z)\sigma^2(x_i,z_j)$ can be expanded into a sum of individual terms. With similar analysis as before, for each individual term in $b_{i,j}^2(x,z)\sigma^2(x_i,z_j)$, the double sum over i,j is either $O\{(nh_n)^{-1}\tilde{\lambda}_1^{-2/m_1}\}$, $O\{(nh_n)^{-1}\tilde{\lambda}_2^{-2/m_2}\}$, or is of smaller order.

Proof of Theorem 2.1: Proposition 2.1 states that the sandwich smoother is asymptotically equivalent to a kernel regression estimator with a product kernel $H_{m_1}(x)H_{m_2}(z)$. To determine the asymptotic bias and variance of the kernel estimator, we conduct a similar analysis of multivariate kernel density estimator as in Wand and Jones (1995). By Proposition 2.1,

$$\mathrm{E}\{\hat{\mu}(x,z)\} = \frac{1}{nh_{n,1}h_{n,2}}\sum_{i,j}\mu(x_i,z_j)H_{m_1}\left(\frac{x-x_i}{h_{n,1}}\right)H_{m_2}\left(\frac{z-z_j}{h_{n,2}}\right) + O(\xi), \quad (2.23)$$

where we continue using the notation $\xi = \max\{(K_1h_{n,1})^{-2}, (K_2h_{n,2})^{-2}\}$. Let

$$\begin{aligned}\mu_0(x,z) = &\frac{1}{nh_{n,1}h_{n,2}}\sum_{i,j}\mu(x_i,z_j)H_{m_1}\left(\frac{x-x_i}{h_{n,1}}\right)H_{m_2}\left(\frac{z-z_j}{h_{n,2}}\right)\\ &-\frac{1}{h_{n,1}h_{n,2}}\iint\mu(u,v)H_{m_1}\left(\frac{x-u}{h_{n,1}}\right)H_{m_2}\left(\frac{z-v}{h_{n,2}}\right)\mathrm{d}u\mathrm{d}v.\end{aligned} \quad (2.24)$$

The first term on the right hand of (2.24) is the Riemann finite sum of $(h_{n,1}h_{n,2})^{-1}\mu(u,v)\ H_{m_1}\{h_{n,1}^{-1}(x-u)\}H_{m_2}\{h_{n,2}^{-1}(z-v)\}$ on the grid while the sec-

ond term is the integral of the same function, and $\mu_0(x,z)$ calculates the difference between the two terms. $\mu_0(x,z)$ is not random and Lemma 2.6 shows that $\mu_0(x,z) = O\left\{\max\left(n_1^{-2}h_{n,1}^{-2}, n_2^{-2}h_{n,2}^{-2}\right)\right\}$. Now (2.23) becomes

$$\begin{aligned}\mathrm{E}\left\{\hat{\mu}(x,z)\right\} &= \frac{1}{h_n}\iint \mu(u,v)H_{m_1}\left(\frac{x-u}{h_{n,1}}\right)H_{m_2}\left(\frac{z-v}{h_{n,2}}\right)\mathrm{d}u\mathrm{d}v + \mu_0(x,z) + O(\xi) \\ &= \iint \mu(x-h_{n,1}u, z-h_{n,2}v)H_{m_1}(u)H_{m_2}(v)\mathrm{d}u\mathrm{d}v + \mu_0(x,z) + O(\xi).\end{aligned} \tag{2.25}$$

For the double integral in (2.25), we first take the Taylor expansion of $\mu(x - h_{n_1}u, z - h_{n_2}v)$ at (x,z) until the $2m_1$th partial derivative with respect to x and the $2m_2$th partial derivative with respect to z, and then we cancel out those integrals that vanish by Lemma A.13 in Section A.9 of Appendix A. It follows that explicit expressions for the asymptotic mean can be attained

$$\begin{aligned}\mathrm{E}\left\{\hat{\mu}(x,z)\right\} - \mu(x,z) = \mu_0(x,z) &+ (-1)^{m_1+1}h_{n,1}^{2m_1}\frac{\partial^{2m_1}}{\partial x^{2m_1}}\mu(x,z) \\ &+ (-1)^{m_2+1}h_{n,2}^{2m_2}\frac{\partial^{2m_2}}{\partial z^{2m_2}}\mu(x,z) + o\left(h_{n,1}^{2m_1}\right) + o\left(h_{n,2}^{2m_2}\right) + O(\xi).\end{aligned}$$

For any two random variables X and Y, if $\mathrm{var}(Y) = o\{\mathrm{var}(X)\}$, then $\mathrm{var}(X+Y) = \mathrm{var}(X) + o\{\mathrm{var}(X)\}$. Hence, by letting $X = \mu^*(x,z)$ and $Y = \hat{\mu}(x,z) - \mu^*(x,z)$, we can obtain by Proposition 2.1 that

$$\mathrm{var}\{\hat{\mu}(x,z)\} = (nh_n)^{-1}\sigma^2(x,z)\int H_{m_1}^2(u)du\int H_{m_2}^2(v)dv + o\{(nh_n)^{-1}\}.$$

To get optimal rate of convergence, let $h_{n,1}^{2m_1}/h_{n,2}^{2m_2}$ and $h_{n,1}^{4m_1}/(nh_n)^{-1}$ converge to some constants, respectively. Then we have

$$h_{n,1} \sim h_1 n^{-m_2/m_3}, h_{n,2} \sim h_2 n^{-m_1/m_3}$$

for some positive constants h_1 and h_2. (Recall that $m_3 = 4m_1m_2 + m_1 + m_2$.) We need to choose K_1, K_2 so that $\max\{(K_1h_{n,1})^{-2}, (K_2h_{n,2})^{-2}\} = o(h_{n,1}^{2m_1})$. Hence, $K_1 \sim C_1 n^{\tau_1}$ for some positive constant C_1 and $\tau_1 > (m_1m_2 + m_2)/m_3$. Similarly,

$K_2 \sim C_2 n^{\tau_2}$ for some positive constant C_2 and $\tau_2 > (m_1 m_2 + m_1)/m_3$. It is easy to verify that $\max\left(n_1^{-2} h_{n,1}^{-2}, n_2^{-2} h_{n,2}^{-2}\right) = o(h_{n,1}^{2m_1})$.

Lemma 2.5 *Let $G(x)$ be a real function in $[0,1]$ with a continuous second derivative. Let $x_i = (i - 1/2)/n$ for $i = 1, \ldots, n$. Assume $h = o(1), (nh^2)^{-1} = o(1)$ as n goes to infinity. Then*

$$\left| \frac{1}{h} \int_0^1 H_m\left(\frac{x-u}{h}\right) G(u) du - \frac{1}{nh} \sum_{i=1}^{n} H_m\left(\frac{x-x_i}{h}\right) G(x_i) \right| = O(n^{-2} h^{-2}),$$

where $H_m(x)$ is defined in (2.14).

Proof of Lemma 2.5: First note that $H_m(x)$ is symmetric and is bounded by 1. Also $H_m(x)$ is infinitely differentiable over $(-\infty, 0]$ and all of its derivatives are bounded by m over $(-\infty, 0]$. Let $L_i = [(i-1)/n, i/n]$ for $i = 1, \ldots, n$. Suppose without loss of generality that $\max_{u \in [0,1]} |G(u)| \le m$. We have

$$\begin{aligned} &\left| \frac{1}{h} \int_0^1 H_m\left(\frac{x-u}{h}\right) G(u) du - \frac{1}{nh} \sum_{i=1}^{n} H_m\left(\frac{x-x_i}{h}\right) G(x_i) \right| \\ \le &\sum_{i=1}^{n} \left| \frac{1}{h} \int_{L_i} \left\{ H_m\left(\frac{x-u}{h}\right) G(u) - H_m\left(\frac{x-x_i}{h}\right) G(x_i) \right\} du \right|, \end{aligned} \tag{2.26}$$

and

$$\begin{aligned} &\left| \frac{1}{h} \int_{L_i} \left\{ H_m\left(\frac{x-u}{h}\right) G(u) - H_m\left(\frac{x-x_i}{h}\right) G(x_i) \right\} du \right| \\ \le &\left| \frac{G(x_i)}{h} \int_{L_i} \left\{ H_m\left(\frac{x-u}{h}\right) - H_m\left(\frac{x-x_i}{h}\right) \right\} du \right| \\ &+ \left| \frac{1}{h} H_m\left(\frac{x-x_i}{h}\right) \int_{L_i} \{G(u) - G(x_i)\} du \right| \\ &+ \left| \frac{1}{h} \int_{L_i} \left\{ H_m\left(\frac{x-u}{h}\right) - H_m\left(\frac{x-x_i}{h}\right) \right\} \{G(u) - G(x_i)\} du \right| \\ \le &\left| \frac{m}{h} \int_{L_i} \left\{ H_m\left(\frac{x-u}{h}\right) - H_m\left(\frac{x-x_i}{h}\right) \right\} du \right| + O(n^{-3} h^{-1}) + O(n^{-3} h^{-2}). \end{aligned} \tag{2.27}$$

In the derivation of (2.27), the term $O(n^{-3}h^{-1})$ follows from

$$\left|G(u) - G(x_i) - (u - x_i)\frac{\partial G}{\partial x}(x_i)\right| \leq \frac{1}{2}(u - x_i)^2 \max_{0\leq x\leq 1}\left|\frac{\partial^2 G}{\partial x^2}(x)\right|$$

and

$$\left|\int_{L_i}\{G(u) - G(x_i)\}\,du\right| = \left|\int_{L_i}\left\{G(u) - G(x_i) - (u - x_i)\frac{\partial G}{\partial x}(x_i)\right\}du\right|;$$

the term $O(n^{-3}h^{-2})$ follows from

$$\left|\frac{1}{h}\left\{H_m\left(\frac{x-u}{h}\right) - H_m\left(\frac{x-x_i}{h}\right)\right\}\{G(u) - G(x_i)\}\right| = O(n^{-2}h^{-2})$$

since $|u - x_i| \leq n^{-1}$ when both u and x_i are in L_i. Note that we used the equality $\int_{L_i}(u - x_i)du = 0$ in the above derivation and we shall use it later as well. Combining (2.26) and (2.27), we have

$$\begin{aligned}&\left|\frac{1}{h}\int_0^1 H_m\left(\frac{x-u}{h}\right)G(u)du - \frac{1}{nh}\sum_{i=1}^n H_m\left(\frac{x-x_i}{h}\right)G(x_i)\right| \\ \leq& m\sum_{i=1}^n\left|\frac{1}{h}\int_{L_i}\left\{H_m\left(\frac{x-u}{h}\right) - H_m\left(\frac{x-x_i}{h}\right)\right\}du\right| + O(n^{-2}h^{-2}).\end{aligned} \tag{2.28}$$

For simplicity, denote $H_m^{(1)}(x), H_m^{(2)}(x)$ the first and second derivatives of $H_m(x)$, respectively. Similarly, denote $H_m^{(1)}(0)$ and $H_m^{(2)}(0)$ the right derivatives of $H_m(x)$ at 0. If $x \in L_i$, then $H_m\{h^{-1}(x-u)\} - H_m\{h^{-1}(x-x_i)\} = O(n^{-1}h^{-1})$ and hence

$$\left|\frac{1}{h}\int_{L_i}\left\{H_m\left(\frac{x-u}{h}\right) - H_m\left(\frac{x-x_i}{h}\right)\right\}du\right| = O(n^{-2}h^{-2}), \text{ if } x \in L_i. \tag{2.29}$$

Assume $x < (i-1)/n$, then $x \notin L_i$. Let

$$\begin{aligned}\tilde{H}_m(u, x_i, x, h) =& H_m\left(\frac{x-u}{h}\right) - H_m\left(\frac{x-x_i}{h}\right) - \frac{u-x_i}{h}H_m^{(1)}\left(\frac{x-x_i}{h}\right) \\ &- \frac{(u-x_i)^2}{2h^2}H_m^{(2)}\left(\frac{x-x_i}{h}\right).\end{aligned}$$

Then $\tilde{H}_m(u, x_i, x, h) = O(h^{-3}|u - x_i|^3)$. We have

$$\begin{aligned}
&\left|\frac{1}{h}\int_{L_i}\left\{H_m\left(\frac{x-u}{h}\right) - H_m\left(\frac{x-x_i}{h}\right)\right\}du\right| \\
=&\left|\frac{1}{h}\int_{L_i}\left\{H_m\left(\frac{x-u}{h}\right) - H_m\left(\frac{x-x_i}{h}\right) - \frac{u-x_i}{h}H_m^{(1)}\left(\frac{x-x_i}{h}\right)\right\}du\right| \\
\leq&\left|\frac{1}{h}\int_{L_i}\frac{(u-x_i)^2}{2h^2}H_m^{(2)}\left(\frac{x-x_i}{h}\right)du\right| + \left|\frac{1}{h}\int_{L_i}\tilde{H}_m(u, x_i, x, h)du\right| \\
\leq&\frac{1}{2n^2h^2}\int_{L_i}\frac{1}{h}\left|H_m^{(2)}\left(\frac{x-x_i}{h}\right)\right|du + O(n^{-4}h^{-4}). \qquad (2.30)
\end{aligned}$$

We can similarly prove that (2.30) holds when $x > i/n$. Now with (2.29) and (2.30),

$$\begin{aligned}
&\sum_{i=1}^{n}\left|\frac{1}{h}\int_{L_i}\left\{H_m\left(\frac{x-u}{h}\right) - H_m\left(\frac{x-x_i}{h}\right)\right\}du\right| \\
\leq&\frac{1}{2n^2h^2}\int_0^1\frac{1}{h}\left|H_m^{(2)}\left(\frac{x-x_i}{h}\right)\right|du + O(n^{-3}h^{-4}) + O(n^{-2}h^{-2}),
\end{aligned}$$

which finishes the lemma.

Lemma 2.6 *The term $\mu_0(x, z)$ defined in (2.24) is $O\left\{\max\left(n_1^{-2}h_{n,1}^{-2}, n_2^{-2}h_{n,2}^{-2}\right)\right\}$.*

Proof of Lemma 2.6: To simplify notation, let $G_2(u, z) = h_{n,2}^{-1}\int_0^1 H_{m_2}\{h_{n,2}^{-1}(z - v)\}\mu(u, v)dv$ and $G_1(u, z) = (n_2h_{n,2})^{-1}\sum_j H_{m_2}\{h_{n,2}^{-1}(z - z_j)\}\mu(u, z_j) - G_2(u, z)$. Then G_1 is $O\{n_2^{-2}h_{n,2}^{-2}\}$ by Lemma 2.5. Note that $|\mu_0(x, z)|$ is bounded by the sum of

$$\left|\frac{1}{n_1h_{n,1}}\sum_i H_{m_1}\left(\frac{x-x_i}{h_{n,1}}\right)G_1(x_i, z)\right| \qquad (2.31)$$

and

$$\left|\frac{1}{n_1h_{n,1}}\sum_j H_{m_1}\left(\frac{x-x_i}{h_{n,1}}\right)G_2(x_i, z) - \frac{1}{h_{n,1}}\int H_{m_1}\left(\frac{x-u}{h_{n,1}}\right)G_2(u, z)du\right|. \qquad (2.32)$$

Because G_1 is $O\left(n_2^{-2}h_{n,2}^{-2}\right)$, (2.31) is also $O\left(n_2^{-2}h_{n,2}^{-2}\right)$. By Theorem 9.1 in the appendix of Durrett (2005), $\partial^2G_2/\partial u^2$ exists and is equal to $h_{n,2}^{-1}\int_0^1 H_{m_2}\{h_{n,2}^{-1}(z -$

$v)\}\partial^2\mu(u,v)/\partial u^2 dv$. Hence $\partial^2 G_2/\partial u^2$ is continuous and bounded. Lemma 2.5 implies (2.32) is $O\left(n_1^{-2}h_{n,1}^{-2}\right)$ which finishes our proof.

Proof of Theorem 2.2: Denote the design points $\{x_i, z_i\}_{i=1}^n$ by $(\underline{\mathrm{x}}, \underline{\mathrm{z}})$. Applying Lemma 2.4 and the proof of Proposition 2.1 to the binned data $\tilde{\mathbf{Y}}$ with n_1, n_2 replaced by I_1, I_2, we obtain

$$\mathrm{E}\left\{\hat{\mu}(x,z)|(\underline{\mathrm{x}}, \underline{\mathrm{z}})\right\} = (Ih_n)^{-1}\sum_{\kappa,\ell}\mathrm{E}\left\{\tilde{y}_{\kappa,\ell}|(\underline{\mathrm{x}}, \underline{\mathrm{z}})\right\}G_{\kappa,\ell}, \tag{2.33}$$

$$\mathrm{var}\left\{\hat{\mu}(x,z)|(\underline{\mathrm{x}}, \underline{\mathrm{z}})\right\} = (Ih_n)^{-2}\sum_{\kappa,\ell}\mathrm{var}\left\{\tilde{y}_{\kappa,\ell}|(\underline{\mathrm{x}}, \underline{\mathrm{z}})\right\}G_{\kappa,\ell}^2, \tag{2.34}$$

where

$$G_{\kappa,\ell} = H_{m_1}\left(\frac{x-\tilde{x}_\kappa}{h_{n,1}}\right)H_{m_2}\left(\frac{z-\tilde{z}_\ell}{h_{n,2}}\right) + b_{\kappa,\ell}(x,z),$$

and $b_{\kappa,\ell}(x,z)$ is defined similarly to $b_{i,j}(x,z)$ in the proof of Proposition 2.1 with also n_1, n_2 replaced by I_1, I_2. Let $n_{\kappa,\ell}$ be the number of data points in the (κ,ℓ)th bin. Then

$$\mathrm{var}\left\{\tilde{y}_{\kappa,\ell}|(\underline{\mathrm{x}}, \underline{\mathrm{z}})\right\} = n_{\kappa,\ell}^{-2}\sum_{i=1}^n \sigma^2(x_i,z_i)\delta_{\{|x_i-\tilde{x}_\kappa|\le(2I_1)^{-1},|z_i-\tilde{z}_\ell|\le(2I_2)^{-1}\}}.$$

So $\mathrm{var}\left\{\sqrt{n_{\kappa,\ell}}\tilde{y}_{\kappa,\ell}|(\underline{\mathrm{x}}, \underline{\mathrm{z}})\right\}$ is a Nadaraya-Watson kernel regression estimator of the conditional variance function $\sigma^2(x,z)$ at $(\tilde{x}_\kappa, \tilde{z}_\ell)$. Similarly, we can show $n_{\kappa,\ell}/(nI^{-1})$ is a kernel density estimator of $f(x,z)$ at $(\tilde{x}_\kappa, \tilde{z}_\ell)$. By the uniform convergence theory for kernel density estimators and Nadaraya-Watson kernel regression estimators (see, for instance, Hansen (2008)),

$$\sup_{\kappa,\ell}\left|n_{\kappa,\ell}/(nI^{-1}) - f(\tilde{x}_\kappa, \tilde{z}_\ell)\right| = O_p\left\{\sqrt{I\ln n/n} + I^{-2}\right\} = o_p(1), \tag{2.35}$$

and

$$\sup_{\kappa,\ell}\left|\mathrm{var}\left\{\sqrt{n_{\kappa,\ell}}\tilde{y}_{\kappa,\ell}|(\underline{\mathrm{x}}, \underline{\mathrm{z}})\right\} - \sigma^2(\tilde{x}_\kappa, \tilde{z}_\ell)\right| = O_p\left\{\sqrt{I\ln n/n} + I^{-2}\right\} = o_p(1).$$

It follows by the above two equalities that

$$\sup_{\kappa,\ell}\left|\frac{n}{I}\text{var}\left\{\tilde{y}_{\kappa,\ell}|(\underline{\mathrm{x}},\underline{\mathrm{z}})\right\}-\frac{\sigma^2(\tilde{x}_\kappa,\tilde{z}_\ell)}{f(\tilde{x}_\kappa,\tilde{z}_\ell)}\right|=o_p(1). \tag{2.36}$$

With similar argument as in the proof of Proposition 2.1, for any continuous function $g(x,z)$ over $[0,1]^2$, we can derive that

$$\frac{1}{Ih_n}\sum_{\kappa,\ell}g(\tilde{x}_\kappa,\tilde{z}_\ell)G_{\kappa,\ell}^2=g(x,z)\int H_{m_1}^2(u)du\int H_{m_2}^2(v)dv+o(1). \tag{2.37}$$

Then by equalities (2.34) and (2.36),

$$\left|\text{var}\left\{\hat{\mu}(x,z)|(\underline{\mathrm{x}},\underline{\mathrm{z}})\right\}-\frac{1}{nh_nIh_n}\sum_{\kappa,\ell}\frac{\sigma^2(\tilde{x}_\kappa,\tilde{z}_\ell)}{f(\tilde{x}_\kappa,\tilde{z}_\ell)}G_{\kappa,\ell}^2\right|=\frac{o_p(1)}{nh_nIh_n}\sum_{\kappa,\ell}G_{\kappa,\ell}^2=o_p\{(nh_n)^{-1}\}. \tag{2.38}$$

By letting $g(x,z)=\sigma^2(x,z)/f(x,z)$ in (2.37), we derive from (2.38) that

$$\text{var}\left\{\hat{\mu}(x,z)|(\underline{\mathrm{x}},\underline{\mathrm{z}})\right\}=\frac{1}{nh_n}\frac{V(x,z)}{f(x,z)}+o_p\{(nh_n)^{-1}\}, \tag{2.39}$$

where $V(x,z)$ is defined in (2.17). We can write $\text{E}\left\{\tilde{y}_{\kappa,\ell}|(\underline{\mathrm{x}},\underline{\mathrm{z}})\right\}$ as

$$\text{E}\left\{\tilde{y}_{\kappa,\ell}|(\underline{\mathrm{x}},\underline{\mathrm{z}})\right\}=(n_{\kappa,\ell})^{-1}\sum_{i=1}^{n}\mu(x_i,z_i)\delta_{\{|x_i-\tilde{x}_\kappa|\le(2I_1)^{-1},|z_i-\tilde{z}_\ell|\le(2I_2)^{-1}\}}.$$

Equality (2.35) implies each bin is nonempty, so by taking a Taylor expansion of $\mu(x_i,z_j)$ at $(\tilde{x}_\kappa,\tilde{z}_\ell)$ we derive from the above equation that

$$\sup_{\kappa,\ell}\left|\text{E}\left\{\tilde{y}_{\kappa,\ell}|(\underline{\mathrm{x}},\underline{\mathrm{z}})\right\}-\mu(\tilde{x}_\kappa,\tilde{z}_\ell)\right|=O_p(I^{-1/2}).$$

It follows by equality (2.33) that

$$\left|\text{E}\left\{\hat{\mu}(x,z)|(\underline{\mathrm{x}},\underline{\mathrm{z}})\right\}-\frac{1}{Ih_n}\sum_{\kappa,\ell}\mu(\tilde{x}_\kappa,\tilde{z}_\ell)G_{\kappa,\ell}\right|=O_p(I^{-1/2})\frac{1}{Ih_n}\sum_{\kappa,\ell}|G_{\kappa,\ell}|=O_p(I^{-1/2}). \tag{2.40}$$

It is easy to derive that

$$\frac{1}{Ih_n}\sum_{\kappa,\ell}\mu(\tilde{x}_\kappa,\tilde{z}_\ell)G_{\kappa,\ell}=\mu(x,z)+n^{-(2m_12m_2)/m_3}\tilde{\mu}(x,z)+o\left\{n^{-(2m_12m_2)/m_3}\right\},$$

where $\tilde{\mu}(x,z)$ is defined in (2.16). In light of equality (2.40) and the assumption that $I \sim c_I n^{\tau}$ with $\tau > (4m_1m_2)/m_3$,

$$\mathrm{E}\left\{\hat{\mu}(x,z)|(\underline{\mathrm{x}},\underline{\mathrm{z}})\right\} = \mu(x,z) + n^{-(2m_1 2m_2)/m_3}\tilde{\mu}(x,z) + o_p\left\{n^{-(2m_1 2m_2)/m_3}\right\}. \quad (2.41)$$

With (A.6) and (2.41), we can derive that

$$n^{(2m_1 2m_2)/m_3}\left[\hat{\mu}(x,z) - \mathrm{E}\left\{\hat{\mu}(x,z)|(\underline{\mathrm{x}},\underline{\mathrm{z}})\right\}\right] \Rightarrow N\left\{0, V(x,z)/f(x,z)\right\} \quad (2.42)$$

in distribution and

$$n^{(2m_1 2m_2)/m_3}\left[\mathrm{E}\left\{\hat{\mu}(x,z)|(\underline{\mathrm{x}},\underline{\mathrm{z}})\right\} - \mu(x,z)\right] = \tilde{\mu}(x,z) + o_p(1). \quad (2.43)$$

Equalities (2.42) and (2.43) together prove the theorem.

CHAPTER 3

FAST COVARIANCE FUNCTION ESTIMATION

3.1 Introduction

This chapter is based on joint work with David Ruppert, Vadim Zipunnikov and Ciprian Crainiceanu.

Covariance function is an important part of functional and longitudinal data analysis. Specific examples include functional principal component analysis (FPCA), functional linear regression, and functional canonical correlation and discriminant analysis; see Diggle *et al.* (1994), Ramsay and Silverman (2002, 2005), and Ferraty and Vieu (2006) for a comprehensive treatment of these subjects. Covariance function is the functional analogue of the variance-covariance matrix in multivariate data analysis and it summarizes the dependency of observations at different time points or locations. Covariance function characterizes some important properties of the sample path such as smoothness (see, e.g., Stein, 1999). Therefore estimation of covariance function is of significant interest.

A naive estimate of the covariance function is the sample covariance function. However, the sample covariance function is often not practically useful. In many applications such as longitudinal studies, it is only possible to observe each random curve at discrete sampling points with measurement errors. In such a setting, the eigenvectors obtained from discretizing the sample covariance function as approximations to the eigenfunctions tend to be wiggly and can be difficult to interpret. Therefore, smoothing is an important step in the estimation of functional principal components, see, e.g., Rice and Silverman (1991), Capra and Müller (1997),

and Cardot (2000). Smoothing the sample covariance function by some bivariate smoothers such as kernel and local polynomials is a popular nonparametric approach for obtaining a smooth estimate of the covariance function; see, e.g., Staniswalis and Lee (1998), Yao *et al.* (2003) and Yao *et al.* (2005). Alternatively, one may take a two-step procedure, where each random curve is first smoothed and then the covariance function is estimated by the sample covariance function of the smoothed curves (Ramsay and Silverman, 2002). We take the first approach and propose to estimate the covariance function by the sandwich smoother introduced in Chapter 2. We are particularly interested in estimating the covariance function for functional data when dense or regular sampling points are observed for each sample. Because of a fast implementation in Section 3.3.3, we name our method the FACE for fast covariance estimation.

Although FACE is based on the sandwich smoother, FACE has several advantages over the sandwich smoother for smoothing high-dimensional covariance operators. First, FACE exploited the *decomposable* structure of covariance operator (defined later) and can be much faster. Second, FACE can scale up linearly with respect to the dimensionality of functional observations (see Proposition 3.3). In particular, FACE can work with partitioned data when only sequential access to data is available. Third, through smoothing FACE provides not only estimates of the covariance function but also simultaneously the associated eigenfunctions. Fourth, FACE provides efficient ways of calculating principal scores which are important ingredients of FPCA.

3.2 Model Settings and Notation

Let $\{X(t), t \in [0,1]\}$ be a stochastic process with associated covariance function $K(s,t) = \text{cov}\{X(s), X(t)\}, s,t \in [0,1]$. For simplicity, we assume $\text{E}X(t) = 0$, $t \in [0,1]$. Then $K(s,t) = \text{cov}\{X(s), X(t)\}$. Suppose $\{X_i(s), i = 1, \cdots, n,\}$ is a collection of independent realizations of the above stochastic process and we observe the random functions X_i at discrete sampling points with measurement errors,

$$Y_{ij} = X_i(t_j) + \epsilon_{ij}, \quad j = 1, \ldots, m, \ i = 1, \ldots, n,$$

where the sampling points $\{t_1, \ldots, t_m\}$ are independently drawn from a common distribution on $[0,1]$ and are the same across the subjects, and ϵ_{ij} are independent and identically distributed measurement errors with mean zero and variance σ^2. In addition, the random functions X, sampling points t, and measurement errors ϵ are assumed to be mutually independent. The case with subject-specific sampling points is discussed later.

The raw covariance can be computed at each pair of sampling points (t_j, t_ℓ) by $\hat{K}(t_j, t_\ell) = n^{-1} \sum_i Y_{ij} Y_{i\ell}$. Define $\widehat{\mathbf{K}} = \{\hat{K}(t_j, t_\ell)\}_{1 \le j,\ell \le m}$ which is called the sample covariance matrix. Note that estimation of the covariance function by bivariate smoothers is generally based on the data $\widehat{\mathbf{K}}$. To simplify notation, let $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{im})^T, i = 1, \ldots, n$. Then $\widehat{\mathbf{K}} = n^{-1} \sum_{i=1}^n \mathbf{Y}_i \mathbf{Y}_i^T = n^{-1} \mathbf{Y}\mathbf{Y}^T$ where $\mathbf{Y} = [\mathbf{Y}_1, \ldots, \mathbf{Y}_n]$ is an $m \times n$ matrix with the ith column $\mathbf{Y}_i$. Because $\widehat{\mathbf{K}}$ can be written as a sum of rank-one matrixes, we call $\widehat{\mathbf{K}}$ has a *decomposable* structure. This *decomposable* structure will be useful in deriving a fast algorithm for computing FACE.

We shall use the big "O" and the small "o" notation. If $a = O(b)$, then

$\limsup |a/b| < \infty$ and if $a = o(b)$, then $\limsup |a/b| = 0$. Furthermore, we use the big Theta notation where $a = \Theta(b)$ means $0 < \limsup |a/b| < \infty$.

3.3 Fast Covariance Function Estimation

The FACE method proposes to smooth the sample covariance matrix using the sandwich smoother so that

$$\widetilde{\mathbf{K}} = \mathbf{S}\widehat{\mathbf{K}}\mathbf{S}, \tag{3.1}$$

where $\widetilde{\mathbf{K}}$ is the smoothed $m \times m$ matrix and $\mathbf{S}$ is a symmetric smoother matrix of dimension $m \times m$. We also use P-splines (Eilers and Marx, 1996) to construct the smoother matrix $\mathbf{S}$ so that $\mathbf{S} = \mathbf{B}\left(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{D}^T\mathbf{D}\right)^{-1}\mathbf{B}^T$, where $\mathbf{B}$ is the $m \times c$ matrix $\{B_k(t_j)\}_{1\le j\le m, 1\le k\le c}$, $\mathbf{D}$ is the differencing matrix of size $(c - m_0) \times c$, and λ is the smoothing parameter. Here $\{B_1(\cdot), \ldots, B_c(\cdot)\}$ is the collection of B-spline basis functions and c is the number of knots plus the degrees of B-splines. Also m_0 is the order of the difference penalty. We assume the knots are equally spaced. Model (3.1) is special case of the sandwich smoother. However FACE has some further important characteristics.

An important characteristic of $\widetilde{\mathbf{K}}$ is that it is guaranteed to be symmetric and positive semi-definite since $\widehat{\mathbf{K}}$ is so. Moreover, the most important practical consequence of the sandwich form of the smoother in (3.1) is that it can be exploited to scale FACE to high and ultra-high dimensional data. None of the current methods is currently designed to handle such covariance operators. Our experience is that smoothing a 500×500 matrix is already a computationally hard task for most bivariate smoothers.

The estimated covariance function is obtained once the smoothing parameter

is selected. Then the eigendecomposition of $\widetilde{\mathbf{K}}$ provides us estimates of the eigenfunctions associated with the covariance function. However, when m is large, both the smoother matrix and the sample covariance matrix will be high-dimensional and can be computationally expensive to calculate, and the eigendecomposition of $\widetilde{\mathbf{K}}$ is also computationally hard. Next we show with a fixed smoothing parameter how to obtain the eigendecomposition of $\widetilde{\mathbf{K}}$ without directly computing the smoother matrix and the sample covariance matrix and without a brute-force eigendecomposition of $\widetilde{\mathbf{K}}$. The derivation will also provide insights on how to select the smoothing parameter efficiently.

3.3.1 Estimation of Eigenfunctions

Assuming that the covariance function K is square integrable in $L_2([0,1]^2)$, Mercer's theorem states that K admits an eigenvalue decomposition $K(s,t) = \sum_k \lambda_k \psi_k(s)\psi_k(t)$ where $\{\psi_k(\cdot) : k \geq 1\}$ is a set of orthonormal basis of $L_2([0,1])$ and $\lambda_1 \geq \lambda_2 \geq \cdots$ are the eigenvalues. Estimating the functional principal components/eigenfunctions ψ_k's is one of the most fundamental tasks in functional data analysis and has attracted a lot of attention in the literature (see, e.g., Ramsay and Silverman, 2005). In particular, the interest lies in seeking the first few eigenfunctions that explain a majority of amount of variation in the observed data, or in other words, finding the first few eigenfunctions such that the random functions X_i can be well approximated by a linear combination of these eigenfunctions. Computing the eigenfunctions of a symmetric bivariate function is generally not trivial. The common practice is to discretize the estimated covariance function and approximate its eigenfunctions by the respective eigenvectors (see, e.g., Rice and Silverman, 1991, Capra and Müller, 1997). In this section, we show that

through smoothing with FACE, we simultaneously obtain the eigendecomposition of the smoothed covariance matrix $\widetilde{\mathbf{K}}$ in (3.1) and thus obtain estimates of the eigenfunctions.

We proceed as in Section 2.2.2 of Chapter 2 with the following spectral decomposition, $(\mathbf{B}^T\mathbf{B})^{-1/2}\mathbf{D}^T\mathbf{D}(\mathbf{B}^T\mathbf{B})^{-1/2} = \mathbf{U}\text{diag}(\mathbf{s})\mathbf{U}^T$, where $\mathbf{U}$ is the matrix of eigenvectors and $\mathbf{s}$ is the vector of eigenvalues. Let $\mathbf{A}_S = \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1/2}\mathbf{U}$. Then $\mathbf{A}_S^T\mathbf{A}_S = \mathbf{I}_c$ which implies $\mathbf{A}_S$ has orthonormal columns. It follows that $\mathbf{S} = \mathbf{A}_S\mathbf{\Sigma}_S\mathbf{A}_S^T$ with $\mathbf{\Sigma}_S = \{\mathbf{I}_c + \lambda\text{diag}(\mathbf{s})\}^{-1}$.

Let $\widetilde{\mathbf{Y}} = \mathbf{A}_S^T\mathbf{Y}$ be a $c \times n$ matrix, then

$$\widetilde{\mathbf{K}} = \mathbf{A}_S\left(\frac{1}{n}\mathbf{\Sigma}_S\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^T\mathbf{\Sigma}_S\right)\mathbf{A}_S^T. \tag{3.2}$$

Equation (3.2) is very important. First, it shows that only the $c \times c$ matrix in the parenthesis depends on the smoothing parameter and this is the bedrock for an algorithm selecting the smoothing parameter efficiently (see Section 3.3.2). Second, it provides the spectral decomposition of $\widetilde{\mathbf{K}}$ with just one more computation. Specifically, suppose we have the spectral decomposition

$$n^{-1}\mathbf{\Sigma}_S\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^T\mathbf{\Sigma}_S = \mathbf{A}\mathbf{\Sigma}\mathbf{A}^T, \tag{3.3}$$

where $\mathbf{A}$ is the $c \times c$ matrix of eigenvectors and $\mathbf{\Sigma}$ is the $c \times c$ diagonal matrix with eigenvalues (This eigendecomposition is needed only after the smoothing parameter has been selected.), then

$$\tilde{\mathbf{K}} = (\mathbf{A}_S\mathbf{A})\mathbf{\Sigma}(\mathbf{A}_S\mathbf{A})^T \tag{3.4}$$

gives the eigendecomposition of $\widetilde{\mathbf{K}}$. Because of the dimension reduction of matrices ($c \times c$ versus $m \times m$), the eigendecomposition in (3.3) is much easier than that of the smoothed covariance matrix. The above derivation shows that through smoothing we obtain not only an smoothed covariance operator but also simultaneously its

associated eigenfunctions. Note that in FPCA, generally we want only the eigen-decomposition of the covariance operator instead of the covariance operator itself. When only a small number of estimated eigenfunctions are desired, the number of elements to store is only $\Theta(m)$ for FACE, while using other bivariate smoothers requires storing the $m \times m$ smoothed covariance operator first. Therefore, FACE is much more flexible in dealing with limited computing resources when m is large. See Proposition 3.3 for an evaluation of the computation time and storage space burden of FACE.

3.3.2 Selection of the Smoothing Parameter

We have the following proposition.

Proposition 3.1 *Assume $c = o(m)$, then the smoothed covariance matrix $\widetilde{\mathbf{K}}$ in (3.2) has a rank at most equal to* $\min(c, n)$.

By Proposition 3.1, the number of knots controls the maximal rank of the smoothed covariance matrix, $\widetilde{\mathbf{K}}$, or equivalently, the number of eigenfunctions that can be extracted from $\widetilde{\mathbf{K}}$. With a small number of knots, we are unable to recover a sufficient number of eigenfunctions. Furthermore, we may have biased estimates of the eigenfunctions because the complex shapes of the eigenfunctions are missed when the spline basis is small. Therefore, we shall use a relatively large number of knots. The overfitting problem induced by the large number of knots can be partially offset by an appropriate penalty. See Ruppert (2002) and Wang *et al.* (2011) for simulations and theory on the knots selection problem. Next we focus on selecting the smoothing parameter.

The penalization induced by the smoothing parameter λ is designed to reduce overfitting of the eigenfunctions when a large number of spline basis functions is used. The effective degrees of freedom of the smoother matrix $\mathbf{S}$, i.e., $tr(\mathbf{S})$, decreases as λ increases. When $\lambda = 0$, the effective degrees of freedom of $\mathbf{S}$ is c and when λ is infinite, the effective degrees of freedom is the degree of the B-splines used for constructing the smoother matrix.

The fast algorithm in Section 2.2.2 for selecting smoothing parameters for the sandwich smoother can be directly applied to FACE. Because FACE uses two identical univariate smoother matrices (see equation (3.1)), there is only one smoothing parameter to select. Hence we can obtain a fast selection of smoothing parameter for FACE. However, when applying the algorithm in Section 2.2.2 of Chapter 2, we can see that we need to calculate $\|\widehat{\mathbf{K}}\|_F^2 = \|n^{-1}\mathbf{Y}^T\mathbf{Y}\|_F^2$, which requires about $\Theta(mn^2)$ computations. So the calculation of $\|\widehat{\mathbf{K}}\|_F^2$ can be expensive when either m or n is large. To overcome this, we propose another approach for selecting the smoothing parameter: select the smoothing parameter by minimizing GCV which is of the form

$$\sum_{i=1}^{n} \|\mathbf{Y}_i - \mathbf{S}\mathbf{Y}_i\|^2 / (1 - \mathrm{tr}(\mathbf{S})/m)^2. \tag{3.5}$$

Here $\|\cdot\|$ is the Euclidean norm of a vector. Regarding each sample as a smoothed curve measured with errors, a number of papers used the above criterion to selecting the smoothing parameter for each curve (Besse and Ramsay, 1986; Ramsay and Dalzell, 1991; Besse, Cardot and Ferraty, 1997). Using the same smoothing parameter for smoothing each sample is reasonable as the samples are i.i.d. of the same stochastic process.

Proposition 3.2 *The GCV in (3.5) equals to*

$$\frac{\sum_{k=1}^{c} C_{kk}(\lambda s_k)^2/(1+\lambda s_k)^2 - \|\widetilde{\mathbf{Y}}\|_F^2 + \|\mathbf{Y}\|_F^2}{\left\{1 - m^{-1}\sum_{k=1}^{c}(1+\lambda s_k)^{-1}\right\}^2},$$

where s_k is the kth element of $\mathbf{s}$*, C_{kk} is the kth diagonal element of* $\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^T$*, and* $\|\cdot\|_F$ *is the Frobenius norm.*

Proposition 3.2 provides an efficient calculation of GCV. We just need to calculate $\|\mathbf{Y}\|_F^2$ and $\|\widetilde{\mathbf{Y}}\|_F^2$ once for computing all GCV. Moreover, we need also only one calculation of the diagonals of $\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^T$. It is easy to see that the calculation of $\|\mathbf{Y}\|_F^2$, $\|\widetilde{\mathbf{Y}}\|_F^2$ and the diagonal of $\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^T$ is only $\Theta(mn + cn)$.

3.3.3 Fast Algorithm

A fast algorithm for FACE:

Step 1 : Specify $\mathbf{S}$ by calculating and storing $\mathbf{A}_S$ and $\mathbf{s}$.

Step 2 : Calculate and store $\widetilde{\mathbf{Y}} = \mathbf{A}_S'\mathbf{Y}$.

Step 3 : Select λ by minimizing GCV in (3.5).

Step 4 : Construct the eigendecomposition in (3.3).

Step 5 : Construct the eigendecomposition in (3.4).

Proposition 3.3 evaluates the computational complexity of FACE using the above algorithm.

Proposition 3.3 *The computation time of FACE is* $\Theta\left(mnc + mc^2 + c^3 + ck\right)$*, where k is the number of iterations needed for selecting the smoothing parameter (see Section 3.3.2), and the total required storage space is* $\Theta\left(mn + n^2 + mc + c^2 + k\right)$ *memory units.*

Remark 3.1 *When $c = \Theta(n)$ and $k = o(mn)$, the computation time of FACE is $\Theta(mn^2 + n^3)$ and the storage space needed is $\Theta(mn + n^2)$ memory units. As a comparison, if we smooth the covariance operator using other bivariate smoothers, then at least $\Theta(m^2 + mn)$ memory units are needed to store the data matrix and the sample covariance matrix and the computational burden can be even more challenging for large m.*

3.3.4 FACE as a Two-step Procedure

The proposed FACE method can be regarded as a two-step procedure (see, e.g., Besse and Ramsay, 1986; Ramsay and Dalzell, 1991; Besse, Cardot and Ferraty, 1997) as follows. First we smooth each functional observation $\mathbf{Y}_i$ so that $\widehat{\mathbf{Y}}_i = \mathbf{S}\mathbf{Y}_i, i = 1, \ldots, n$. Then we construct a smooth covariance operator based on the smoothed functional observations $\mathbf{Y}_i$'s using method of moments. It is easy to show we also obtain model (3.1). The FACE method is computationally advantageous to a general two-step procedure. There are two computational novelties in the FACE method. First, FACE selects the smoothing parameter using all functional observations while in a general two-step procedure the smoothing parameters are subject-specific. The fast algorithm in Section 3.3.2 enables FACE to select efficiently the smoothing parameter.

Second, FACE is a convenient method because it provides not only a smoothed covariance operator but also simultaneously the eigendecomposition as shown in Section 3.3.1. With a general two-step procedure that smoothes each functional observation separately, we obtain smoothed functional observations of length m. We then may need to obtain the eigendecomposition of the sample covariance matrix based on the smoothed functional observations.

3.3.5 Subject-specific Sampling Points

Previous discussions assume the sampling points are not subject-specific. Assume now for the ith sample, we observe $\mathbf{Y}_i = \{Y_i(t_{i1}), \ldots, Y_i(t_{im_i})\}^T$ which implies the ith sample has m_i data points. Because the sampling points differ across the subjects, we do not have an estimated covariance operator that takes the simple form $n^{-1}\sum_{i=1}^{n} \mathbf{Y}_i\mathbf{Y}_i^T$.

We consider the scenario that each subject is densely sampled, i.e., all m_i's are large. Extending the idea in Di *et al.* (2008), we can use a kernel smoother with a very small bandwidth or a regression spline smoother to each $\mathbf{Y}_i$ to obtain an under-smoothed estimate, $\widehat{\mathbf{Y}}_i$, at a finite grid. We can then apply FACE to the set of under-smoothed estimates, $\{\widehat{\mathbf{Y}}_1, \ldots, \widehat{\mathbf{Y}}_n\}$.

For the case with sparse sampling points for the observations, our method does not apply as an implicit assumption is that each subject is densely sampled. One may use the common bivariate smoothers such as the local polynomials to estimate the covariance function.

3.3.6 Estimation of Principal Scores in FPCA

Under some regularity condition on the sample path (Karhunen, 1947), X_i can be written as $X_i(t) = \sum_{k\geq 1} \xi_{ik}\psi_k(t)$ where $\{\psi_k : k \geq 1\}$ is the set of eigenfunctions of K and $\xi_{ik} = \int_0^1 X_i(s)\psi_k(s)ds$ are the principals scores of X_i. It follows that

$$Y_{ij} = \sum_{k\geq 1} \xi_{ik}\psi_k(t_j) + \epsilon_{ij}.$$

In practice, we may be only interested in the first N eigenfunctions of the covariance operator and hence approximately,

$$Y_{ij} = \sum_{k=1}^{N} \xi_{ik}\psi_k(t_j) + \epsilon_{ij}.$$

With estimated eigenfunctions $\hat{\psi}_k$'s and estimated eigenvalues $\hat{\lambda}_k$'s from FACE, we can obtain the principal scores of each X_i either through numerical integration or as BLUPs (best unbiased linear predictors). Note that the $\hat{\lambda}_k$'s are the eigenvalues of $\widetilde{\mathbf{K}}$ divided by m. Next we show that FACE provides fast calculations of scores for both approaches.

Let $\widetilde{\mathbf{Y}}_i$ denote the ith column of $\widetilde{\mathbf{Y}}$. Let $\boldsymbol{\xi}_i = (\xi_{i1}, \ldots, \xi_{iN})^T$ and let $\widehat{\mathbf{A}}_N$ denote the first N columns of $\mathbf{A}$ defined in (3.3). Let $\boldsymbol{\psi}_k = \{\psi_k(t_1), \ldots, \psi_k(t_m)\}^T$ and $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_N]$. The matrix $m^{-1/2}\boldsymbol{\Psi}$ is estimated by $\mathbf{A}_S\widehat{\mathbf{A}}_N$ by FACE. For the method of numerical integration, we have

$$\hat{\xi}_{ik} = \int_0^1 X_i(s)\hat{\psi}_k(s)ds \approx m^{-1}\sum_{i=1}^{m} X_i(t_j)\hat{\psi}_k(t_j). \tag{3.6}$$

Theorem 3.1 *The estimated principal scores* $\hat{\boldsymbol{\xi}}_i = (\hat{\xi}_{i1}, \ldots, \hat{\xi}_{iN})^T$ *obtained from (3.6) are given by*

$$\hat{\boldsymbol{\xi}}_i = m^{-1/2}\widehat{\mathbf{A}}_N^T\widetilde{\mathbf{Y}}_i, \quad 1 \le i \le n.$$

Although in general estimated BLUPs are better than estimates from numerical integration as the former have smaller variances, the differences between those two estimates are little when m is large. For completeness, below we provide the estimated BLUPs. Let $\epsilon_{ij} = Y_{ij} - \sum_{k=1}^{N}\psi_k(t_j)\xi_{ik}$ and $\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \ldots, \epsilon_{im})^T$. Then $\mathbf{Y}_i = \boldsymbol{\Psi}\boldsymbol{\xi}_i + \boldsymbol{\epsilon}_i$. The covariance matrix $\mathrm{var}(\boldsymbol{\xi}_i) = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ is estimated by $m^{-1}\widehat{\boldsymbol{\Sigma}}_N = m^{-1}\mathrm{diag}(\hat{\lambda}_1, \ldots, \hat{\lambda}_N)$. The variance of ϵ_{ij} can be estimated by

$$\hat{\sigma}^2 = m^{-1}tr\left\{n^{-1}\mathbf{Y}\mathbf{Y}^T - m(\mathbf{A}_S\widehat{\mathbf{A}}_N)\widehat{\boldsymbol{\Sigma}}_N(\mathbf{A}_S\widehat{\mathbf{A}}_N)^T\right\},$$

or equivalently,

$$\hat{\sigma}^2 = m^{-1}n^{-1}\|\mathbf{Y}\|_F^2 - \sum_{k=1}^{N}\hat{\lambda}_k. \tag{3.7}$$

Theorem 3.2 *Suppose* $\boldsymbol{\Psi}$ *is estimated by* $\sqrt{m}\mathbf{A}_S\widehat{\mathbf{A}}_N$, $\mathrm{var}(\boldsymbol{\xi}_i) = diag(\lambda_1, \ldots, \lambda_N)$ *is estimated by* $\widehat{\boldsymbol{\Sigma}}_N = diag(\hat{\lambda}_1, \ldots, \hat{\lambda}_N)$, *and* σ^2 *is estimated by* $\hat{\sigma}^2$ *in (3.7). The estimated BLUPs of* $\hat{\boldsymbol{\xi}}_i$ *are given by*

$$\hat{\boldsymbol{\xi}}_i = m^{-1/2}\widehat{\boldsymbol{\Sigma}}_N\left(\widehat{\boldsymbol{\Sigma}}_N + m^{-1}\hat{\sigma}^2\mathbf{I}_N\right)^{-1}\widehat{\mathbf{A}}_N^T\widetilde{\mathbf{Y}}_i, \quad 1 \le i \le n.$$

Theorems 3.1 and 3.2 provide fast approaches for calculating the principal scores using either numerical integration or BLUPS. These approaches combined with FACE are much faster because they make use of the calculations already done for estimating the eigenfunctions and eigenvalues.

3.4 FACE for Large m or/and Large n

In this section, we adapt FACE to either the large m or the large m and large n setting. FACE is computationally feasible for either of the two settings as we show it can work with partitioned data when the data are massive.

3.4.1 The Case of Large m

In this subsection, we assume $n = o(m)$ and m is large enough to make loading objects of size m in the computer memory impractical. We need to reconsider some steps of our fast algorithm to ensure the scalability of the approach.

In Step 1, we need to reconsider the calculation of $\mathbf{B}^T\mathbf{B}$ and $\mathbf{A}_S$. The matrix $\mathbf{B}$ is of dimension $m \times c$ and hence is too big to be loaded into memory. Following Zipunnikov *et al.* (2011), we partition $\mathbf{B}$ into M small matrices of size $m/M \times c$, i.e., $\mathbf{B}^T = [\mathbf{B}_{(1)}^T, \ldots, \mathbf{B}_{(M)}^T]$. Then $\mathbf{B}^T\mathbf{B} = \sum_{i=1}^M \mathbf{B}_{(i)}^T\mathbf{B}_{(i)}$. We choose M large enough so that the size of $\mathbf{B}_{(i)}$ can be comfortably loaded into memory. The following facts are useful for reducing the computation time: most of the columns of $\mathbf{B}$ has a pattern and most elements of $\mathbf{B}$ are zero because B-splines are piecewise polynomials with local supports. However, we do not discuss the reduction of computation time here. Similarly, we partition $\mathbf{A}_S$ the same way as $\mathbf{B}$, i.e., $\mathbf{A}_S^T = [\mathbf{A}_{S,(1)}^T, \ldots, \mathbf{A}_{S,(M)}^T]$. Then $\mathbf{A}_{S,(i)} = \mathbf{B}_{(i)}(\mathbf{B}^T\mathbf{B})^{-1/2}\mathbf{U}$. So $\mathbf{A}_S$ can be easily computed and stored piecewise.

In Step 2, we need to reconsider the calculation of $\widetilde{\mathbf{Y}} = \mathbf{A}_S^T\mathbf{Y}$. We partition $\mathbf{X}$ similarly so that $\mathbf{X}^T = [\mathbf{X}_{(1)}^T, \ldots, \mathbf{X}_{(M)}^T]$. Then $\widetilde{\mathbf{Y}} = \sum_{i=1}^M \mathbf{A}_{S,(i)}^T\mathbf{Y}_{(i)}$.

In Step 3, we need to reconsider the calculation of $\|\mathbf{Y}\|_F^2$. We have $\|\mathbf{Y}\|_F^2 = \sum_{i=1}^M \|\mathbf{Y}_{(i)}\|_F^2$.

In Step 4, the eigendecomposition in (3.3) does not involve any object of size m.

In Step 5, we need to reconsider the calculation of $\mathbf{A}_S\mathbf{A}$ in the eigendecomposition in (3.4). The above data partition of $\mathbf{A}_S$ can be used and we have $(\mathbf{A}_S\mathbf{A})^T = [\mathbf{A}^T\mathbf{A}_{S,(1)}^T, \ldots, \mathbf{A}^T\mathbf{A}_{S,(M)}^T]$.

3.4.2 The Case of Large m and Large n

We assume either $m = O(n)$ or $m = o(n)$. We assume $c = o\{\min(m, n)\}$. The data partition strategy in Section 3.4.1 can be applied here. Moreover, we

need to reconsider the calculations that involve matrices with n as one dimension. From Section 3.4.1, we need only to reconsider the calculations involving $\mathbf{X}$ and $\widetilde{\mathbf{Y}}$ in Steps 2 and 3 of our algorithm. First, we consider the calculation of $\widetilde{\mathbf{Y}} = \mathbf{A}_S^T\mathbf{X}$ in Step 2. In Section 3.4.1, we partition $\mathbf{Y}$ into M pieces so that $\mathbf{Y}^T = [\mathbf{Y}_{(1)}, \ldots, \mathbf{Y}_{(M)}]$ and $\mathbf{Y}_{(i)}$ is of dimension $m/M \times n$. Now we partition each $\mathbf{Y}_{(i)}$ into M smaller matrices so that $\mathbf{Y}_{(i)} = [\mathbf{Y}_{(i,1)}, \ldots, \mathbf{Y}_{(i,M)}]$. One may choose a different M for partitioning $\mathbf{Y}_{(i)}$. For simplicity, we use the same M. It follows that $\widetilde{\mathbf{Y}} = [\sum_{i=1}^M \mathbf{A}_{S,(i)}^T\mathbf{X}_{(i,1)}, \ldots, \sum_{i=1}^M \mathbf{A}_{S,(i)}^T\mathbf{X}_{(i,M)}]$. Hence we can partition $\widetilde{\mathbf{Y}}$ into $\widetilde{\mathbf{Y}} = [\widetilde{\mathbf{Y}}_{(1)}, \ldots, \widetilde{\mathbf{Y}}_{(M)}]$ with $\widetilde{\mathbf{Y}}_{(j)} = \sum_{i=1}^M \mathbf{A}_{S,(i)}^T\mathbf{Y}_{(i,j)}$. Next, we consider the calculation of the diagonals of $\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^T$ and $\|\mathbf{Y}\|_F^2$ in Step 3. It is easy to show that $\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^T = \sum_{i=1}^M \widetilde{\mathbf{Y}}_{(i)}\widetilde{\mathbf{Y}}_{(i)}^T$ and $\|\mathbf{Y}\|_F^2 = \sum_{i=1}^M \sum_{j=1}^M \|\mathbf{Y}_{(i,j)}\|_F^2$.

3.5 Simulation

We first introduce a simple approach for estimating the eigenfunctions and eigenvalues when the functional data are observed without any noise, i.e., a computationally easier approach for decomposing $\widehat{\mathbf{K}}$. Suppose without loss of generality that $m > n$. Consider the singular value decomposition of $\mathbf{Y}$

$$\mathbf{Y} = \mathbf{U}_y\mathbf{D}_y\mathbf{V}_y^T,$$

where $\mathbf{U}_y$ is an $m \times n$ matrix with orthonormal columns, $\mathbf{V}_y$ is an $n \times n$ matrix with orthonormal columns, and $\mathbf{D}_y$ is an $n \times n$ diagonal matrix. It is easy to see that the columns of $\mathbf{U}_y$ contain all the eigenvectors of $\widehat{\mathbf{K}}$ with non-zero eigenvalues and the set of diagonal elements of $n^{-1}\mathbf{D}_y^2$ have all the non-zero eigenvalues of $\widehat{\mathbf{K}}$. Hence obtaining $\mathbf{U}_y$ and $\mathbf{D}_y$ is equivalent to the eigendecomposition of $\widehat{\mathbf{K}}$. $\mathbf{D}_y$ and $\mathbf{V}_y$ can be computed by $\mathbf{U}_y = \mathbf{Y}\mathbf{V}_y\mathbf{D}_y^{-1}$. We name the above approach the SVD method

and it can be shown that SVD requires $\Theta\{(m+n)\min(n^2, m^2)\}$ computations. By Remark 3.1 we see that when $c = \Theta(n)$, FACE is computationally comparable to SVD.

We illustrate our FACE method with two simulation studies. In the first simulation study, we consider moderately high dimensional data contaminated with noises. We let $m = 3,000$. We show that the estimated eigenfunctions by FACE are more accurate and smooth than these by SVD. Moreover, we show that FACE provides closes estimates of the eigenvalues than SVD. Because of the large m, we do not evaluate other bivariate smoothers which are generally much slower and require much more memory space. (See Remark 3.1 of Section 3.3.)

In the second simulation study, to show that FACE scales up well with high-dimensional data, we assess the computational time of FACE and compare it with that of SVD and the sandwich smoother. The code were written in R and all simulation studies were run on a duo core 2.4 GHz Mac with 4GB of RAM memory.

3.5.1 Covariance Function Estimation

We generate the data from the model

$$\begin{gathered} X_i(t_j) = \sum_{k=1}^{N} \xi_{ik}\psi_k(t_j) + \epsilon_{ij}, \quad 1 \le i \le n, 1 \le j \le m, \\ \xi_{ik} \overset{\text{i.i.d.}}{\sim} N(0, \lambda_k), \quad \epsilon_{ij} \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2), \end{gathered} \tag{3.8}$$

where ξ_{ik}'s and ϵ_{ij}'s are mutually independent. We let $n = 1000$, and let the number of eigenfunctions $N = 4$. The true eigenvalues are $\lambda_k = 0.5^{k-1}, k =$

$1, 2, 3, 4$. We consider two different sets of bases

$$\text{Case 1:} \quad \left\{\sqrt{2}\sin(2\pi t), \sqrt{2}\cos(2\pi t), \sqrt{2}\sin(4\pi t), \sqrt{2}\cos(4\pi t)\right\},$$
$$\text{Case 2:} \quad \left\{1, \sqrt{3}(2t-1), \sqrt{5}(6t^2-6t+1), \sqrt{7}(20t^3-30t^2+12t-1)\right\},$$

which are measured on a regular grid of m equidistant points in the unit interval, $\{1/m, 2/m, \ldots, 1\}$. The above two sets of bases were used in Di *et al.* (2009), Greven *et al.* (2010), and Zipunnikov *et al.* (2011). We let $\sigma = 2$.

Throughout all simulation studies, we use cubic B-splines and a difference penalty of order 2 to construct the univariate smoother matrix. Figures 3.1 is for case 1 and it displays the true and estimated eigenfunctions using SVD, FACE with $c = 500$, and FACE with $c = 1000$ from the top row to the bottom row, respectively. Figure 3.2 is for case 2. Shown in the two figures are the true eigenfunctions (solid red lines), the pointwise median of estimated eigenvectors (dashed black lines) and the pointwise 5th and 95th percentiles of the estimated eigenfunctions.

We see that the estimated eigenfunctions by FACE are smooth and accurate, while it is surprising to see that the estimates given by SVD are also fairly accurate even though they are more wiggly and have more variations from estimates to estimates. Figure 3.3 shows boxplots of estimated eigenvalues that are centered and standardized, $(\hat{\lambda}_k - \lambda_k)/\lambda_k$. On average, SVD slightly overestimates the eigenvalues especially small eigenvalues while FACE is better than SVD for estimating small eigenvalues. It is well known that sample eigenvalues and sample eigenvectors of large covariance matrix with relatively small sample size are inconsistent estimators (see, e.g., Johnstone, 2001; Baik and Silverstein, 2005; Johnstone and Lu, 2009). However, our simulation seems to suggest that sample eigenvalues and sample eigenvectors are consistent for covariance function estimation. Further

investigation of the above problem is beyond the scope of this work.

Figure 3.4 illustrates the fast approach for calculating principal scores in Section 3.3.6.

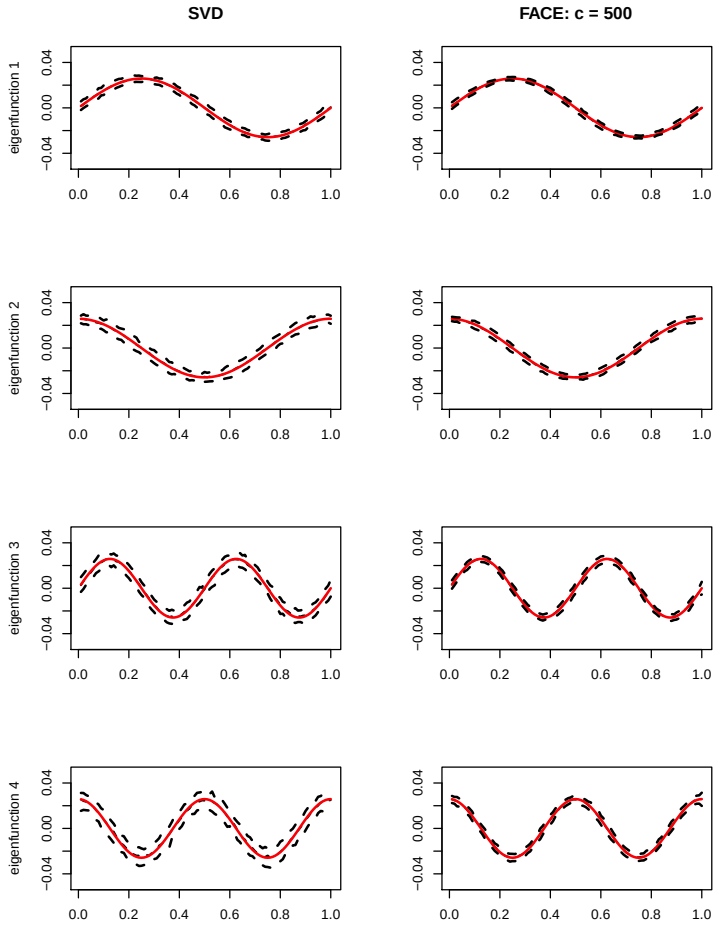


Figure 3.1: True and estimated eigenfunctions of ψ_k's for case 1 replicated 100 times with noises. The variance of noises is 4. Each box shows the true eigenfunction (solid red lines), the pointwise median and the 5th and 95th point wise percentile curves (dashed black lines).

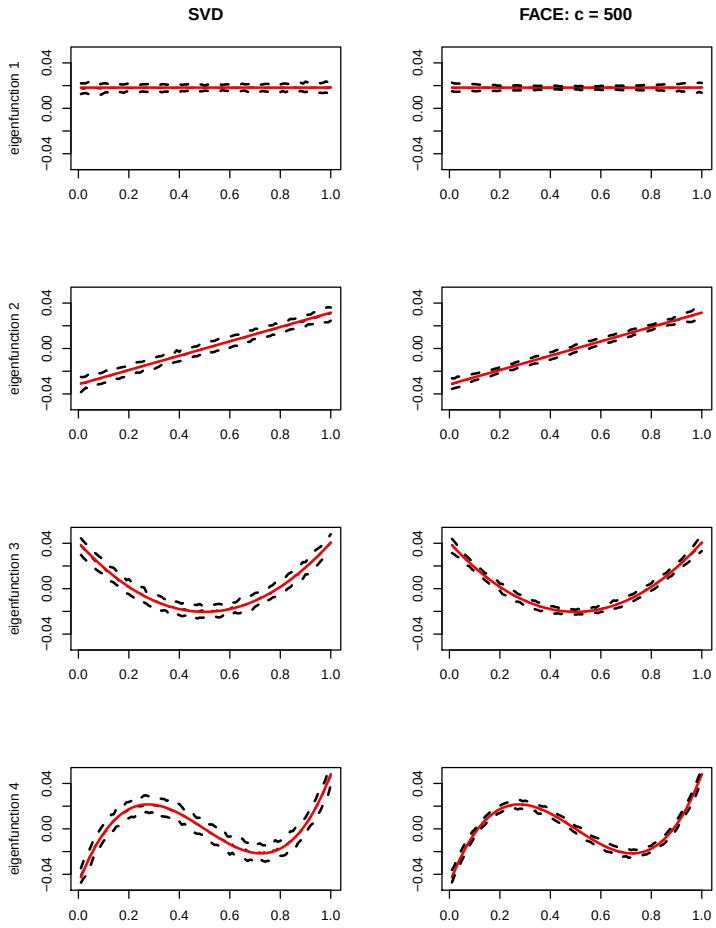


Figure 3.2: True and estimated eigenfunctions of ψ_k's for case 2 replicated 100 times with noises. The variance of noises is 4. Each box shows the true eigenfunction (solid red lines), the pointwise median and the 5th and 95th point wise percentile curves (dashed black lines).

3.5.2 Computation Time

Proposition 3.1 provides theoretic evaluation of the computation speed of FACE, here we record the computation time of FACE for various combinations of m and n. All other settings remain the same as in the first simulation study and we use the eigenfunctions from case 1. As a comparison, the computation time of SVD and the sandwich smoother is also provided. Table 3.1 summarizes the results and

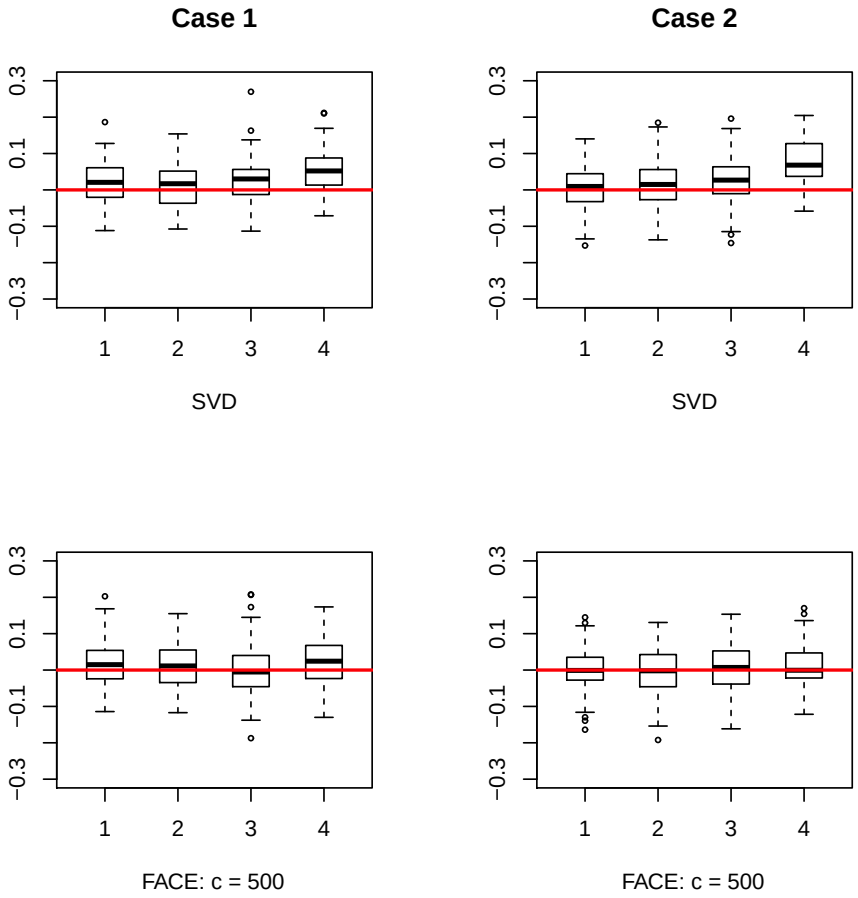


Figure 3.3: Boxplots of the centered and standardized estimated eigenvalues, $(\hat{\lambda}_k - \lambda_k)/\lambda_k$. The left panel is for case 1 and the right panel is for case 2. The zero is shown by the solid red line.

shows that FACE can be quite fast even with high-dimensional data and it can be much faster than the sandwich smoother.

Although we do not run FACE on extremely high-dimensional data for which the data partition strategy in Section 3.4 is needed, we can use Proposition 3.1 to obtain a rough estimate of the computation time. Table 3.1 shows that FACE with $c = 500$ takes 25 seconds on data with $(m, n) = (5,000, 2,000)$. For data with m equal to 0.1 million and n is 20,000, FACE with $c = 500$ should take less than 1.5 hours to compute, without taking into account the time for loading data into

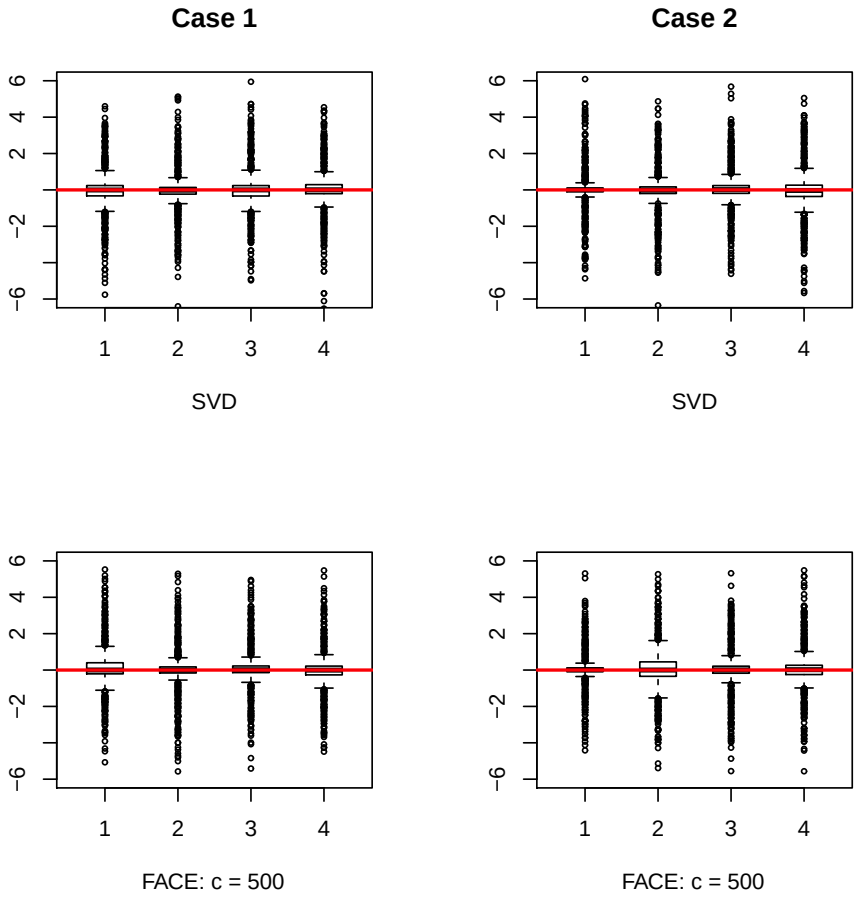


Figure 3.4: Boxplots of the principal scores for the four eigenfunctions. The left panel is for case 1 and the right panel is for case 2. The zero is shown by the solid red line.

the computer memory. Our code is in R, so a faster implementation of FACE can be expected if we use Matlab.

3.6 Proof of Theorems

Proof of Proposition 3.1: The design matrix $\mathbf{B}$ is of full rank. Hence $\mathbf{B}^T\mathbf{B}$ is invertible and $\mathbf{A}_S$ is of rank c. $\mathbf{\Sigma}_S$ is a diagonal matrix with all elements greater

Table 3.1: Computation time (in seconds) of SVD, the sandwich smoother and FACE averaged over 100 data sets on 2.4GHz computers running mac with 4GB of RAM. The number of knots is 500 for both the sandwich smoother and FACE.

m	n	SVD	Sandwich smoother	FACE
3,000	1,000	11.3	130.2	11.7
	2,000	61.1	152.2	15.6
5,000	1,000	19.4	565.5	16.9
	2,000	126.1	645.1	25.2

than 0. $\widetilde{\mathbf{Y}}$ is of rank at most $\min(c, n)$. Hence the matrix in the parenthesis of (3.1) has a rank at most $\min(c, n)$ and the proposition follows.

Proof of Proposition 3.2: First of all, $tr(\mathbf{S}) = tr(\mathbf{\Sigma}_S)$ which is easy to calculate. We now compute $\sum_{i=1}^n \|\mathbf{Y}_i - \mathbf{S}\mathbf{Y}_i\|^2$. Because $\|\mathbf{Y}_i - \mathbf{S}\mathbf{Y}_i\|^2 = \mathbf{Y}_i^T(\mathbf{S} - \mathbf{I}_m)^2\mathbf{Y}_i = tr\{(\mathbf{S} - \mathbf{I}_m)^2\mathbf{Y}_i\mathbf{Y}_i^T\}$ where $(\mathbf{S} - \mathbf{I}_m)^2$ is well defined as $\mathbf{S} - \mathbf{I}_m$ is symmetric,

$$\sum_{i=1}^n \|\mathbf{Y}_i - \mathbf{S}\mathbf{Y}_i\|^2 = tr\left\{(\mathbf{S} - \mathbf{I}_m)^2 \sum_{i=1}^n \mathbf{Y}_i\mathbf{Y}_i^T\right\} = tr\left\{(\mathbf{S} - \mathbf{I}_m)^2\mathbf{Y}\mathbf{Y}^T\right\}.$$

It can be shown that $\mathbf{S}^2 = \mathbf{A}_S\mathbf{\Sigma}_S^2\mathbf{A}_S^T$. Hence $tr(\mathbf{S}^2\mathbf{Y}\mathbf{Y}^T) = tr(\mathbf{Y}^T\mathbf{S}^2\mathbf{Y}) = tr(\widetilde{\mathbf{Y}}^T\mathbf{\Sigma}_S^2\widetilde{\mathbf{Y}}) = tr(\mathbf{\Sigma}_S^2\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^T)$. Similarly, we derive $tr(\mathbf{S}\mathbf{Y}\mathbf{Y}^T) = tr(\mathbf{\Sigma}_S\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^T)$. We have $tr(\mathbf{Y}\mathbf{Y}^T) = \|\mathbf{Y}\|_F^2$. It follows that

$$\sum_{i=1}^n \|\mathbf{Y}_i - \mathbf{S}\mathbf{Y}_i\|^2 = tr\left\{(\mathbf{\Sigma}_S - \mathbf{I}_c)^2\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^T\right\} - \|\widetilde{\mathbf{Y}}\|_F^2 + \|\mathbf{Y}\|_F^2,$$

and

$$tr\left\{(\mathbf{\Sigma}_S - \mathbf{I}_c)^2\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^T\right\} = \sum_{k=1}^c (\lambda s_k)^2/(1 + \lambda s_k)^2 C_{kk}.$$

Proof of Proposition 3.3: We need to compute or store $\mathbf{Y}$, $\mathbf{B}$, $\mathbf{B}^T\mathbf{B}$, $(\mathbf{B}^T\mathbf{B})^{-1/2}$, $\mathbf{D}^T\mathbf{D}$, $(\mathbf{B}^T\mathbf{B})^{-1/2}\mathbf{D}^T\mathbf{D}(\mathbf{B}^T\mathbf{B})^{-1/2}$, $\mathbf{A}_S$, $\widetilde{\mathbf{Y}}$, $\mathbf{A}$, $\mathbf{U}$, and $\mathbf{A}_S\mathbf{A}$. For the computational complexity, $\mathbf{B}^T\mathbf{B}$, $\mathbf{A}_S = \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1/2}\mathbf{U}$, and $\mathbf{A}_S\mathbf{A}$ require $\Theta(mc^2)$ computations; $(\mathbf{B}^T\mathbf{B})^{-1/2}$, $\mathbf{D}^T\mathbf{D}$, $(\mathbf{B}^T\mathbf{B})^{-1/2}\mathbf{D}^T\mathbf{D}(\mathbf{B}^T\mathbf{B})^{-1/2}$, $\mathbf{A}$, and $\mathbf{U}$ require $\Theta(c^3)$ computations; $\widetilde{\mathbf{Y}} = \mathbf{A}_S^T\mathbf{Y}$ requires $\Theta(mnc)$ computations. So in total, $\Theta(mpnc + mc^2 + c^3)$ computations are required. For the memory burden, the loading of $\mathbf{X}$ requires $\Theta(mn)$ memory units, storage of $\mathbf{B}$ and $\mathbf{A}_S\mathbf{A}$ requires $\Theta(mc)$ memory units, and other objects require $\Theta(c^2)$ memory units.

Proof of Theorem 3.1: By (3.6), $\hat{\boldsymbol{\xi}}_i = p^{-1/2}(\mathbf{A}_S\widehat{\mathbf{A}}_N)^T\mathbf{Y}_i = p^{-1/2}\widehat{\mathbf{A}}_N^T(\mathbf{A}_S^T\mathbf{Y}_i) = p^{-1/2}\widehat{\mathbf{A}}_N^T\widetilde{\mathbf{Y}}_i$.

Proof of Theorem 3.2: Let $\widetilde{\mathbf{A}}_N$ denote the first N columns of $\mathbf{A}_S\mathbf{A}$, then $\widetilde{\mathbf{A}}_N = \mathbf{A}_S\widehat{\mathbf{A}}_N$. The estimated BLUPs for $\boldsymbol{\xi}_i$ (Ruppert, *et al.*, 2003) is

$$\hat{\boldsymbol{\xi}}_i = m^{1/2}\widehat{\boldsymbol{\Sigma}}_N\widetilde{\mathbf{A}}_N^T\left(m\widetilde{\mathbf{A}}_N\widehat{\boldsymbol{\Sigma}}_N\widetilde{\mathbf{A}}_N^T + \hat{\sigma}^2\mathbf{I}_m\right)^{-1}\mathbf{Y}_i.$$

The inverse matrix in the above equality can be replaced by the following (Seber, 2007, pp. 309),

$$\left(m\widehat{\mathbf{A}}_N\widehat{\boldsymbol{\Sigma}}_N\widetilde{\mathbf{A}}_N^T + \hat{\sigma}^2\mathbf{I}_m\right)^{-1} = \frac{1}{\hat{\sigma}^2}\left\{\mathbf{I}_m - \frac{m}{\hat{\sigma}^2}\widetilde{\mathbf{A}}_N\widehat{\boldsymbol{\Sigma}}_N\left(\mathbf{I}_N + \frac{m}{\hat{\sigma}^2}\widehat{\boldsymbol{\Sigma}}_N\right)^{-1}\widetilde{\mathbf{A}}_N^T\right\}.$$

It follows that

$$\begin{aligned}
\hat{\boldsymbol{\xi}}_i &= m^{1/2}\frac{1}{\hat{\sigma}^2}\widehat{\boldsymbol{\Sigma}}_N\left\{\mathbf{I}_N - \frac{m}{\hat{\sigma}^2}\widehat{\boldsymbol{\Sigma}}_N\left(\mathbf{I}_N + \frac{m}{\hat{\sigma}^2}\widehat{\boldsymbol{\Sigma}}_N\right)^{-1}\right\}\widetilde{\mathbf{A}}_N^T\mathbf{Y}_i \\
&= m^{-1/2}\widehat{\boldsymbol{\Sigma}}_N(\widehat{\boldsymbol{\Sigma}}_N + m^{-1}\hat{\sigma}^2\mathbf{I}_N)^{-1}\widetilde{\mathbf{A}}_N^T\mathbf{Y}_i \\
&= m^{-1/2}\widehat{\boldsymbol{\Sigma}}_N(\widehat{\boldsymbol{\Sigma}}_N + m^{-1}\hat{\sigma}^2\mathbf{I}_N)^{-1}\widehat{\mathbf{A}}_N\widetilde{\mathbf{Y}}_i,
\end{aligned}$$

which proves the theorem.

APPENDIX A

LOCAL ASYMPTOTICS OF P-SPLINES

A.1 Introduction

This appendix is based on joint work with Yingxing Li, Tatiyana V. Apanasovich and David Ruppert.

Suppose there is a univariate regression model

$$y_i = \mu(x_i) + \epsilon_i, \quad i = 1, \ldots, n,$$

where $\mu(x_i)$ and $\sigma^2(x_i)$ are the conditional expectation and variance of y_i given x_i, respectively. For simplicity, we assume $x_i \in [0, 1]$.

The regression function $\mu(x)$ can be modeled by $\sum_{k=1}^{c} \theta_k B_k(x)$ where $c = K + p$ and $\mathbf{B}(x) = \{B_1(x), \ldots, B_c(x)\}^T$ is a B-spline basis of degree p with knots $0 = \kappa_0 < \kappa_1 < \cdots < \kappa_K = 1$. P-splines (Eilers and Marx, 1996) find $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \ldots, \hat{\theta}_c)^T$ that minimizes

$$\sum_{i=1}^{n} \left\{ y_i - \sum_{k=1}^{c} \hat{\theta}_k B_k(x_i) \right\}^2 + \lambda^* \sum_{k=m+1}^{c} \left\{ \Delta^n \left(\hat{\theta}_k \right) \right\}^2, \quad \lambda \geq 0, \tag{A.1}$$

where Δ is the difference operator, i.e., $\Delta(\theta_k) = \theta_k - \theta_{k-1}$ and $\Delta^m = \Delta(\Delta^{m-1})$, and λ^* is the smoothing parameter. Minimizing (A.1) gives

$$\left(\mathbf{B}^T \mathbf{B}/M + \lambda \mathbf{D}^T \mathbf{D} \right) \hat{\boldsymbol{\theta}} = \mathbf{B}^T \mathbf{y}, \tag{A.2}$$

where $M = n/K$, $\lambda = \lambda^* K/n$, $\mathbf{y} = (y_1, \ldots, y_n)^T$, $\mathbf{B} = \{\mathbf{B}(x_1)^T, \ldots, \mathbf{B}(x_n)^T\}^T$ is an $n \times c$ matrix, and $\mathbf{D}$ is the mth order differencing matrix of dimension $(c-m) \times c$. For simplicity of notation, let

$$\boldsymbol{\Lambda} = \mathbf{B}^T \mathbf{B}/M + \lambda \mathbf{D}^T \mathbf{D} \tag{A.3}$$

which is the smoother matrix for P-splines. Then the estimate is given by

$$\hat{\mu}(x) = \mathbf{B}^T(x)\hat{\boldsymbol{\theta}} = \mathbf{B}^T(x)\boldsymbol{\Lambda}^{-1}\mathbf{B}^T\mathbf{y}/M. \tag{A.4}$$

For simplicity, we assume $x_1 = 1/(2n), x_2 = 3/(2n), \ldots, x_n = (2n-1)/(2n)$, i.e., the response is observed at equally spaced design points. We also assume M is an integer to simplify some proofs. The case when the fixed design points are not equally spaced is considered in Section A.6.

A.2 Review of Theoretical Study

Penalized splines have been popular in recent years, as penalized splines use fewer knots, thus need less computation than smoothing splines. Ruppert *et al.* (2003) treat penalized splines extensively and also give numerous applications.

However, the theory of penalized splines has been remaining an interesting but challenging problem. Opsomer and Hall (2005) first studied the asymptotic theory of penalized splines when K, the number of knots, is infinite. Li and Ruppert (2008) derived the first asymptotic distribution with low degree of splines and with low order of penalty. Wang *et al.* (2009) related penalized splines with some ordinary differential equations (ODEs), and by studying Green's functions associated with those ODEs, they were able to derive the asymptotic distribution of penalized splines.

In contrast to Li and Ruppert (2008), Kauermann *et al.* (2009) considered the situation when K increases at a slower rate. Though they did not obtain an explicit expression for the asymptotic bias and variance, they generalized their results for non-normal responses. Claeskens *et al.* (2009) showed that depending on whether

$K \to \infty$ increasing at a sufficiently fast or a sufficiently slow rate, the asymptotic distribution of penalized splines is either close to that of a smoothing spline or a regression spline. Correspondingly, they referred to these two cases as either a large or small K scenario. The large K scenario is closest to current practice, as discussed, for example, in O'Sullivan (1986), Eilers and Marx (1996), and Ruppert *et al.* (2003), a relatively large number of knots is used and overfitting is controlled by a careful choice of smoothing parameter.

One general approach to the theory of penalized splines is to use an equivalent kernel method, which was first used by Silverman (1984) for studying the asymptotics of smoothing splines. The equivalent kernel method was also useful in studying the asymptotics of P-splines (Li and Ruppert, 2008; Wang *et al.*, 2009).

Independent from Wang *et al.* (2009), we extend Li and Ruppert's (2008) results and provide an explicit expression on the asymptotic distribution of P-splines at an interior point. We also derive the asymptotic distribution of P-splines near the boundary, acknowledging the existence of Wang *et al.* (2009). The conjecture, that provided it is fast enough, the divergence rate of the number of knots does not affect the asymptotic distribution of penalized splines, is confirmed in this paper.

The remainder of this chapter is organized as follows. In Section A.3, we summarize our main results. In Section A.4, we provide a general introduction of our method and present some technical results. In Section A.5, We prove the main results in Section A.3. In Section A.6, we consider irregularly spaced data. In Section A.7, we give an example illustrating the idea of binning data for irregularly space data. In Section A.8, we conclude this chapter with some discussion.

A.3 Main Results

In this section, we summarize the main results. All derivations and proofs are given in Sections A.4 and A.5. For notational convenience, $a \sim b$ implies a/b converges to 1. We use the big "O" and small "o" notation that is with respect to n. Throughout this chapter, $a = O(b)$ means $|a/b|$ converges to some finite nonnegative number as n goes to infinity and $a = o(b)$ mean $|a/b|$ converges to 0. We also denote by $\mu^{(k)}(x)$ the kth derivative of the function $\mu(x)$. We need the following definition.

Definition A.1 *We define a kernel function*

$$H_m(x) = \frac{1}{2m} \sum_{\nu=1}^{m} \psi_\nu \exp\left(-\psi_\nu |x|\right),$$

where $\psi_1, \cdots, \psi_m$ are the m complex roots of $x^{2m} + (-1)^m = 0$ such that all $\psi_\nu (1 \leq \nu \leq m)$ have positive real parts.

A kernel estimator with the kernel H_m is of the form $(nh_n)^{-1} \sum_i y_i H_m\{h_n^{-1}(x - x_i)\}$, where h_n is the bandwidth. As shown in Lemma A.13, H_m is of order $2m$ which determines the convergence rate the corresponding kernel estimator. Proposition A.1 shows that the P-spline estimator at an interior point is asymptotically equivalent to the above kernel estimator.

Proposition A.1 *Assume the following conditions are satisfied.*

1. *There exists a constant $\delta > 0$ such that $\sup_i \mathrm{E}\left(|y_i|^{2+\delta}\right) < \infty$.*
2. *The regression function $\mu(x)$ has a continuous $2m$th order derivative.*
3. *The variance function $\sigma^2(x)$ is continuous.*

4. *The random errors* $\epsilon_i, 1 \leq i \leq n$, *are mutually independent.*

5. *The covariates satisfy* $x_i = (i - 1/2)/n$, $1 \leq i \leq n$.

Let $\psi_0 = \min\{Re(\psi_1), \ldots, Re(\psi_m)\}$, *where* $Re(\cdot)$ *gives the real part of a complex number. Let* $h_n = \lambda^{1/(2m)}/K$. *Assume* $h_n = o(1)$ *and* $(Kh_n)^{-1} = o(1)$. *Let* $\hat{\mu}(x)$ *be the P-spline estimator using* m*th order difference penalty and* p *degree B-splines with equally spaced knots. Fix* $x \in (0, 1)$. *Let* $\mu^*(x) = (nh_n)^{-1} \sum_i y_i H_m\{h_n^{-1}(x - x_i)\}$. *Then*

$$\mathrm{E}\{\hat{\mu}(x) - \mu^*(x)\} = O\left\{(Kh_n)^{-2}\right\},$$
$$\mathrm{var}\{\hat{\mu}(x) - \mu^*(x)\} = o\left\{(nh_n)^{-1}\right\}.$$

Theorem A.1 *Use the same notation in Proposition A.1 and assume all conditions and assumptions there are satisfied. Suppose that* $K \sim Cn^{\tau}$ *with* $\tau > (m + 1)/(4m + 1)$, $h_n \sim hn^{-1/(4m+1)}$ *for positive constants* C *and* h *and* $\lambda \sim (Kh_n)^{2m}$. *For any* $x \in (0, 1)$, *we have that*

$$n^{2m/(4m+1)} \{\hat{\mu}(x) - \mu(x)\} \Rightarrow N\{\tilde{\mu}(x), V(x)\}$$

in distribution as $n \to \infty$, *where*

$$\tilde{\mu}(x) = (-1)^{m+1} h^{2m} \mu^{(2m)}(x), \tag{A.5}$$
$$V(x) = \sigma^2(x) \int H_m^2(u) \mathrm{d}u. \tag{A.6}$$

Remark A.1 *Stone (1980) gave the optimal rates of convergence for nonparametric estimators. For a univariate smooth function* $\mu(x)$ *with a continuous* $2m$*th derivative, the corresponding optimal rate of convergence for estimating* $\mu(x)$ *at any interior point is* $n^{-2m/(4m+1)}$. *Hence the P-spline estimator achieves the optimal rate of convergence.*

Theorem A.2 *Assume conditions (1), (3), (4) and (5) in Proposition A.1 hold. Assume $\mu(x)$ has a continuous mth derivative over $[0,1]$. Suppose that $K \sim Cn^{\tau}$ with $\tau > (m+1)/(2m+1)$, $h_n \sim hn^{-1/(2m+1)}$ for positive constants C and h and $\lambda \sim (Kh_n)^{2m}$. Let $\hat{\mu}(x)$ be the penalized estimator with mth order difference penalty and $p \geq 1$ degree B-splines with equally spaced knots. Assume $x \sim c_x h_n$ where c_x is a constant. Then we have that*

$$n^{m/(2m+1)} \{\hat{\mu}(x) - \mu(x)\} \Rightarrow N\{\tilde{\mu}_0(x), V_0(x)\}$$

in distribution as $n \to \infty$, where

$$\tilde{\mu}_0(x) = (-1)^m h^m \mu^{(m)}(0) \int_{-\infty}^{c_x} u^m \{H_m(u) + H_{b,m}(c_x, c_x - u)\} \, \mathrm{d}u,$$
$$V_0(x) = \sigma^2(0) \int_{-\infty}^{c_x} \{H_m(u) + H_{b,m}(c_x, c_x - u)\}^2 \, \mathrm{d}u.$$

Here $H_{b,m}$ is defined in (A.50).

Remark A.2 *Theorems A.1 and A.2 show that the P-spline smoother has a slower rate of convergence at the boundary than in the interior.*

A.4 Preliminary Derivation

We consider the large K scenario (Claeskens *et al.*, 2009) and assume K and the smoothing parameter λ increase with n at certain rates specified later, respectively.

The matrix $\mathbf{\Lambda}$ in (A.3) is a symmetric and banded matrix. For $q \leq k \leq c - q$ with $q = \max(p, m)$, the kth column of $\mathbf{\Lambda}$ (denoted by $\mathbf{\Lambda}_k$) is

$$(0, \ldots, 0, \omega_q, \ldots, \omega_1, \omega_0, \omega_1, \ldots, \omega_q, 0, \ldots, 0)^T$$

with the kth element being ω_0. We need the following equation

$$\omega_q + \omega_{q-1}\rho + \cdots + \omega_1\rho^{q-1} + \omega_0\rho^q + \omega_1\rho^{q+1} + \cdots + \omega_q\rho^{2q} = 0. \tag{A.7}$$

Equation (A.7) has a compact form

$$\lambda(-1)^m(1-\rho)^{2m}\rho^{q-m} + \rho^{q-p}P(\rho) = 0, \tag{A.8}$$

where

$$P(x) = u_p + u_{p-1}x + \cdots + u_0x^p + u_1x^{p+1} + \cdots + u_px^{2p} \tag{A.9}$$

with the kth column of $\mathbf{B}^T\mathbf{B}$ being

$$(0, \ldots, 0, u_p, \ldots, u_1, u_0, u_1, \ldots, u_p, 0, \ldots, 0)^T. \tag{A.10}$$

Let $\{\rho_\nu, \nu = 1, \ldots, q\}$ be the q roots of (A.8) such that when λ is large, the real parts of the first m roots are all positive and less or equal than 1 and moreover if $p > m$, the other $q - m$ roots converge to zero. Define

$$\mathbf{S}_k = \sum_{\nu=1}^{q} a_\nu \mathbf{T}_k(\rho_\nu), \tag{A.11}$$

where

$$\mathbf{T}_k(\rho) = (\rho^{k-1}, \cdots, \rho, 1, \rho, \cdots, \rho^{c-k})^T. \tag{A.12}$$

For $1 \le \nu \le q$ and $2q \le k \le c - 2q$, it can be shown that $\mathbf{T}_i(\rho_\nu)$ is orthogonal to all columns of $\mathbf{\Lambda}$ except the first q columns, the last q columns and the jth column with $|k - j| < q$. The coefficient vector $\mathbf{a} = (a_1, \ldots, a_q)^T$ can be chosen so that $\mathbf{S}_k$ is orthogonal to all columns of $\mathbf{\Lambda}$ except the kth column, the first q columns and the last q columns. It shall be shown later in this section that $\mathbf{a}$ does not depend on k. Specifically, we find a unique $\mathbf{a}$ such that

$$\mathbf{S}_k^T\mathbf{\Lambda}_k = 1 \quad \text{and} \quad \mathbf{S}_k^T\mathbf{\Lambda}_j = 0, \quad 0 < |k - j| \le q - 1, \tag{A.13}$$

where $\mathbf{\Lambda}_k$ is the kth column of $\mathbf{\Lambda}$ as before.

Fix $x \in (0,1)$. By (A.4), we need only to consider non-zero $B_k(x)$. Hence we assume $k \in (Kx-p-1, Kx+p+1)$. By (A.13) and the definition of $\mathbf{S}_k$, there exists a constant $C>0$ such that,

$$\mathbf{S}_k^T \mathbf{\Lambda}_j = O\left[\exp\left\{-C\lambda^{-1/(2m)} K \min(x,1-x)\right\}\right], \quad 1 \le j \le q, \text{and } c-q \le j \le c. \tag{A.14}$$

Let $\mathbf{e}_k$ be a vector of length c with the kth entry 1 and other elements 0. Define $\tilde{\theta}_k = (\mathbf{S}_k^T \mathbf{\Lambda})\hat{\boldsymbol{\theta}}$. Equation (A.2) implies $\tilde{\theta}_k = \mathbf{S}_k^T \mathbf{B}^T \mathbf{y}$. By (A.13), (A.14) and Lemma A.1, $\tilde{\theta}_k - \hat{\theta}_k = (\mathbf{S}_k^T \mathbf{\Lambda} - \mathbf{e}_k^T)\hat{\boldsymbol{\theta}} = \sum_{i=1}^n \tilde{b}_{i,k} y_i$, where $\tilde{b}_{i,k} = O\left[\exp\left\{-C\lambda^{-1/(2m)} K \min(x,1-x)\right\}\right]$. Let $S_{k,r}$ be the kth element of $\mathbf{S}_k$. By (A.4),

$$\begin{aligned}
\hat{\mu}(x) &= \sum_{k=1}^{c} B_k(x) \mathbf{S}_k^T \mathbf{B}^T \mathbf{y} + \sum_{k=1}^{c} B_k(x)(\tilde{\theta}_k - \hat{\theta}_k) \\
&= \sum_{k=1}^{c} \left[B_k(x) \left\{ \sum_{r=1}^{c} S_{k,r} \sum_{i=1}^{n} B_r(x_i) y_i \right\} \right] + \sum_{|k-Kx| \le p} B_k(x) \left(\sum_{i=1}^{n} \tilde{b}_{i,k} y_i \right) \\
&= \sum_{i=1}^{n} y_i \left\{ \sum_{k,r} B_k(x) B_r(x_i) S_{k,r} + b_i(x) \right\},
\end{aligned} \tag{A.15}$$

where $b_i(x) = \sum_{|k-Kx| \le p} B_k(x) \tilde{b}_{i,k} = O\left[\exp\left\{-C\lambda^{-1/(2m)} K \min(x,1-x)\right\}\right]$. We assume appropriate regularity conditions on the data $\mathbf{y}$ so that interchanging sums in (A.15) is valid. Note that $\sum_{k,r} B_k(x) B_r(x_i) S_{k,r} + b_i(x)$ in (A.15) is the weight of the ith observation for estimating $\hat{\mu}(x)$.

For the boundary case, assume x goes to 0 at a rate of $\lambda^{1/(2m)}/K$, i.e., $x \sim c_x \lambda^{1/(2m)}/K$, where c_x is a constant. We assume that $\lambda^{1/(2m)}/K$ converges to 0. Assume $k \in (Kx-p-1, Kx+p+1)$, then $\mathbf{S}_k$ is orthogonal to all columns of $\mathbf{\Lambda}$ except the kth, the first q and the last q columns. Furthermore, $\mathbf{T}_1(\rho)$ defined in (A.12) can be shown orthogonal to all columns of $\mathbf{\Lambda}$ except the first q and the last q columns. Define $\mathbf{R}_k = \sum_{\nu=1}^{q} \tilde{a}_{k,\nu} \mathbf{T}_1(\rho_\nu)$. Then $\mathbf{S}_k + \mathbf{R}_k$ is orthogonal to all columns of $\mathbf{\Lambda}$ except the kth, the first q and the last q columns for arbitrary

coefficient vector $\tilde{\mathbf{a}}_k = \{\tilde{a}_{k,1}, \ldots, \tilde{a}_{k,q}\}^T$. We find the coefficient vector $\tilde{\mathbf{a}}_k$ so that $\mathbf{S}_k + \mathbf{R}_k$ is orthogonal to all columns of $\mathbf{\Lambda}$ except the kth and the last q columns. Specifically, we find $\tilde{\mathbf{a}}$ such that

$$(\mathbf{S}_k + \mathbf{R}_k)^T \mathbf{\Lambda}_k = 1 \quad \text{and} \quad (\mathbf{S}_k + \mathbf{R}_k)^T \mathbf{\Lambda}_j = 0, \quad 0 < j \leq c - q. \tag{A.16}$$

Then there exists a constant $C_0 > 0$ such that for $c - q \leq j \leq c$, $(\mathbf{S}_k + \mathbf{R}_k)^T \mathbf{\Lambda}_j = O\left[\exp\left\{-C_0 \lambda^{-1/(2m)} K\right\}\right]$. We can derive that, similar to (A.15),

$$\hat{\mu}(x) = \sum_{i=1}^{n} y_i \left\{ \sum_{k,r} B_k(x) B_r(x_i)(S_{k,r} + R_{k,r}) + b_{i,0}(x) \right\}, \tag{A.17}$$

where $R_{k,r}$ is the rth element of $\mathbf{R}_k$ with $R_{k,r} = \sum_{\nu=1}^{q} \tilde{a}_{k,\nu} \rho_\nu^{r-1}$, and $b_{i,0}(x) = O\left[\exp\left\{-C_0 \lambda^{-1/(2m)} K\right\}\right]$.

In the next subsections, we shall derive the coefficients ρ_ν, a_ν and $\tilde{a}_{k,\nu}$.

A.4.1 Derivation of ρ_ν

The case $p \leq m$

In this case $q = m$. Equation (A.8) becomes

$$\lambda(-1)^m (1 - \rho)^{2m} + \rho^{m-p} P(\rho) = 0 \tag{A.18}$$

and $\rho_1, \ldots, \rho_m$ are the m complex roots of (A.18) such that the real part of ρ_ν is positive and less or equal than 1. Proposition A.2 below shows that ρ_ν exists and has an explicit form.

Proposition A.2 *As $\lambda \to \infty$, the roots of equation (A.18) take the following forms*

$$\rho_\nu = 1 - \psi_\nu \lambda^{-1/(2m)} + 1/2 \psi_\nu^2 \lambda^{-1/m} + O\left\{\lambda^{-3/(2m)}\right\}, \quad 1 \leq \nu \leq 2m, \tag{A.19}$$

where $\psi_1, \cdots, \psi_{2m}$ *are the roots of* $x^{2m} + (-1)^m = 0$.

Remark A.3 *To be consistent with the definition in Section A.3, we assume for the first* m *roots,* ψ_ν *have positive real parts and for the last* m *roots,* ψ_ν *have negative real parts. The real parts of* $\rho_1, \ldots, \rho_m$ *are hence positive and equal or less than 1.*

Proof of Proposition A.2: The existence of $2m$ roots for equation (A.18) is obvious from complex analysis. Suppose $1 - \delta_1$ is a root of equation (A.18). Then

$$G_{1,\lambda}(\delta_1) = \lambda(-1)^m \delta_1^{2m} + (1 - \delta_1)^{m-p} P(1 - \delta_1) = 0.$$

Because the leading coefficient for the polynomial $G_{1,\lambda}(\delta_1)$ is $\lambda(-1)^m$ (or $\lambda(-1)^m + \omega_0$ if $m = p$), it is easy to see that δ_1 is uniformly bounded as $\lambda \to \infty$. Hence $(1 - \delta_1)^{m-p} P(1 - \delta_1)$ is uniformly bounded, which implies $\lambda(-1)^m \delta_1^{2m}$ is uniformly bounded. It follows that $\lim_{\lambda\to\infty} \delta_1 = 0$. Then

$$\lim_{\lambda\to\infty} G_{1,\lambda}(\delta_1) = \lim_{\lambda\to\infty} \lambda(-1)^m \delta_1^{2m} + 1 = 0,$$

which implies

$$\delta_1 = \psi_\nu \lambda^{-1/(2m)}(1 + \delta_2), \tag{A.20}$$

where ψ_ν is a root of $x^{2m} + (-1)^m = 0$ for some ν and $\lim_{\lambda\to\infty} \delta_2 = 0$. Substituting (A.20) into $G_{1,\lambda}$ (denoted by $G_{2,\lambda}(\delta_2)$) gives

$$0 = G_{2,\lambda}(\delta_2) = -(1+\delta_2)^{2m} + \left\{1 - \psi_\nu \lambda^{-1/(2m)}(1 + \delta_2)\right\}^{m-p} P\left\{1 - \psi_\nu \lambda^{-1/(2m)}(1 + \delta_2)\right\}. \tag{A.21}$$

It is easy to show that

$$\left\{1 - \psi_\nu \lambda^{-1/(2m)}(1 + \delta_2)\right\}^{m-p} = 1 - (m - p)\psi_\nu \lambda^{-1/(2m)} + o\left\{\lambda^{-1/(2m)}\right\}, \tag{A.22}$$

$$P\left\{1 - \psi_\nu \lambda^{-1/(2m)}(1 + \delta_2)\right\} = P(1) - P'(1)\psi_\nu \lambda^{-1/(2m)} + o\left\{\lambda^{-1/(2m)}\right\}. \tag{A.23}$$

Equalities (A.21)–(A.23), as well as Lemma A.5, imply

$$\delta_2 = \frac{p-m-P'(1)}{2m}\psi_\nu \lambda^{-1/(2m)}(1+\delta_3) = -\frac{1}{2}\psi_\nu \lambda^{-1/(2m)}(1+\delta_3),$$

where $\lim_{\lambda\to\infty}\delta_3 = 0$. By similar analysis, we can show that $\delta_3 = O\left\{\lambda^{-3/(2m)}\right\}$. Hence a root of equation (A.18) takes the form

$$1 - \psi_\nu \lambda^{-1/(2m)} + 1/2\psi_\nu^2 \lambda^{-1/m} + O\{\lambda^{-3/(2m)}\}, \quad \text{for some } \nu.$$

Thus, equation (A.18) has $2m$ roots that take the above form and each root has a ψ_ν that is a root of (A.19).

The case $p > m$

When $p > m$, equation (A.8) becomes

$$\lambda(-1)^m(1-\rho)^{2m}\rho^{p-m} + P(\rho) = 0. \tag{A.24}$$

Similar to Proposition A.2, we have the following

Proposition A.3 *As $\lambda \to \infty$, $2m$ roots of equation (A.24) take the forms in (A.19), and additionally, $p-m$ roots of equation (A.24) take the following forms*

$$\rho_\nu = \left\{\frac{\omega_q}{\lambda}\right\}^{\frac{1}{p-m}} \psi_\nu + O(\lambda^{-\frac{2}{p-m}}), \quad m+1 \le \nu \le p, \tag{A.25}$$

where $\psi_{m+1}, \cdots, \psi_p$ are the roots of $x^{p-m} + (-1)^m = 0$.

Proof of Proposition A.3: Assume δ_0 is a root of equation (A.25). Consider the case $\limsup_{\lambda\to\infty}\delta_0 \neq 0$ and is bounded. Then a similar proof as that of Proposition A.2 gives $2m$ roots taking the forms in (A.19). Now consider the case $\limsup_{\lambda\to\infty}\delta_0 = 0$. $P(\delta_0)$ converges to ω_q as $\lambda \to \infty$, which implies $\lambda(-1)^m\delta_0^{p-m}$

converges to $-\omega_q$. It follows that $\delta_0 = \psi_\nu(\omega_q/\lambda)^{1/(p-m)}(1+\delta_1)$, where ψ_ν is a root of $x^{p-m} + (-1)^m = 0$ for some ν and $\lim_{\lambda\to\infty}\delta_1 = 0$. Similar derivation as in the proof of Proposition A.2 gives (A.25). To complete the proof, notice that for the case $\limsup_{\lambda\to\infty}\delta_0 = \infty$, we can derive the rest $p-m$ unbounded roots of equation (A.24).

A.4.2 Derivation of a_ν

In this subsection, we shall establish the following

Proposition A.4 *Assume $q < k < c - q$ and $x \in (0,1)$. As $\lambda \to \infty$, the vector* **a** *satisfying the constraints in (A.13) is unique, i.e., does not depend on k, and has the following form*

$$a_\nu = \frac{\psi_\nu}{2m}\lambda^{-1/(2m)}\left\{1 + O(\lambda^{-1/m})\right\}, \quad 1 \le \nu \le m, \tag{A.26}$$

and if $p > m$,

$$a_\nu = O\left\{\lambda^{p/(m-p)}\right\}, \quad \nu = m+1, \ldots, p.$$

Remark A.4 *Because the proof is lengthy, we shall sketch the proof within the context in the remainder of this subsection.*

For $1 \leq \nu \leq q$, define $s_j(\rho_\nu) = \mathbf{T}_k^T(\rho_\nu)\mathbf{\Lambda}_{i-q+j}$ for $1 \leq j \leq q$. Then $s_j(\rho_\nu) = \sum_{l=0}^{j-1} \omega_{q-l}(\rho_\nu^{j-l} - \rho_\nu^{l-j})$. Constraints in (A.13) give a system of linear equations

$$\begin{pmatrix} s_1(\rho_1) & \cdots & s_1(\rho_q) \\ \vdots & \ddots & \vdots \\ s_{q-1}(\rho_1) & \cdots & s_{q-1}(\rho_q) \\ s_q(\rho_1) & \cdots & s_q(\rho_q) \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_{q-1} \\ a_q \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}.$$

As shall be shown soon, a_ν's exist and are unique. Making use of the structure of $s_j(\rho_\nu)$ and doing row transforms on the above linear equations, we have

$$\begin{pmatrix} \omega_q(\rho_1 - \rho_1^{-1}) & \cdots & \omega_q(\rho_q - \rho_q^{-1}) \\ \vdots & \ddots & \vdots \\ \omega_q(\rho_1^{q-1} - \rho_1^{1-q}) & \cdots & \omega_q(\rho_q^{q-1} - \rho_q^{1-q}) \\ \omega_q(\rho_1^{q} - \rho_1^{-q}) & \cdots & \omega_q(\rho_q^{q} - \rho_q^{-q}) \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_{q-1} \\ a_q \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}.$$

Further row transforms on the above equations give

$$\begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ (\rho_1 + \rho_1^{-1} - 2)^{q-2} & \cdots & (\rho_q + \rho_q^{-1} - 2)^{q-2} \\ (\rho_1 + \rho_1^{-1} - 2)^{q-1} & \cdots & (\rho_q + \rho_q^{-1} - 2)^{q-1} \end{pmatrix} \begin{pmatrix} a_1(\rho_1 - \rho_1^{-1}) \\ \vdots \\ a_{q-1}(\rho_{q-1} - \rho_{q-1}^{-1}) \\ a_q(\rho_q - \rho_q^{-1}) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \omega_q^{-1} \end{pmatrix}.$$

In the above equations, the matrix before the column of coefficients is a $q \times q$ Vandermonde matrix. Making use of the determinant property of Vandermonde matrix, the solution to the above linear equations exists and is unique because

$\rho_\nu + \rho_\nu^{-1} - 2, 1 \le \nu \le q$ are all different. Furthermore, it is apparent that the solution to the above equations does not depend on k, hence $\mathbf{a}$ is the same for all k such that $q \le k \le c-q$. By Cramer's rule in solving linear equations, we obtain for $1 \le \nu \le q$

$$\begin{aligned} a_\nu \omega_q(\rho_\nu - 1/\rho_\nu) =& \frac{(-1)^{m+\nu} \prod_{1\le i<j\le q, j\ne\nu, i\ne\nu}(\rho_j + \rho_j^{-1} - \rho_i - \rho_i^{-1})}{\prod_{1\le i<j\le q}(\rho_j + \rho_j^{-1} - \rho_i - \rho_i^{-1})} \\ =& \frac{(-1)^{q+\nu}(-1)^{q-\nu}}{\prod_{1\le j\ne\nu\le q}(\rho_\nu + \rho_\nu^{-1} - \rho_j - \rho_j^{-1})} \\ =& \frac{1}{\prod_{1\le j\ne\nu\le q}(\rho_\nu + \rho_\nu^{-1} - \rho_j - \rho_j^{-1})}. \end{aligned} \tag{A.27}$$

Hence

$$a_\nu^{-1} = \omega_q(\rho_\nu - \rho_\nu^{-1}) \prod_{1\le j\ne\nu\le q} (\rho_\nu + \rho_\nu^{-1} - \rho_j - \rho_j^{-1}). \tag{A.28}$$

The case $p \le m$

By (A.19), for $1 \le \nu \le m$,

$$\rho_\nu - \rho_\nu^{-1} = -2\psi_\nu \lambda^{-1/2m} + O(\lambda^{-3/2m}),$$

and

$$\rho_\nu + \rho_\nu^{-1} - 2 = \psi_\nu^2 \lambda^{-1/m} + O(\lambda^{-2/m}).$$

It follows that for $1 \le j \ne \nu \le m$,

$$\rho_\nu + \rho_\nu^{-1} - \rho_j - \rho_j^{-1} = (\psi_\nu^2 - \psi_j^2)\lambda^{-1/m} + O(\lambda^{-2/m}). \tag{A.29}$$

Then

$$\begin{aligned} \prod_{j\ne\nu}(\rho_\nu + \rho_\nu^{-1} - \rho_j - \rho_j^{-1}) =& \lambda^{-1+1/m} \prod_{j\ne\nu} \left\{(\psi_\nu^2 - \psi_j^2) + O(\lambda^{-1/m})\right\} \\ =& \lambda^{-1+1/m} \left\{\prod_{j\ne\nu}(\psi_\nu^2 - \psi_j^2) + O(\lambda^{-1/m})\right\}. \end{aligned} \tag{A.30}$$

By Lemma A.6, equality (A.30) can be simplified

$$\prod_{j\neq\nu}(\rho_\nu+\rho_\nu^{-1}-\rho_j-\rho_j^{-1})=(-1)^{m+1}m\psi_\nu^{-2}\lambda^{-1+1/m}\{1+O(\lambda^{-1/m})\}. \tag{A.31}$$

In light of (A.27) and (A.31),

$$\left\{a_\nu\omega_q(\rho_\nu-\rho_\nu^{-1})\right\}^{-1}=(-1)^{m+1}m^{-1}\psi_\nu^2\lambda^{1-1/m}\{1+O(\lambda^{-1/m})\}.$$

Note that for $p\leq m$, $\omega_q=\omega_m=(-1)^m\lambda+$ a constant, where the constant is the coefficient of ρ^m in the polynomial $P(\rho)$. Hence $(-1)^m\lambda^{-1}\omega_q=1+O(\lambda^{-1})$. It follows that

$$\begin{aligned}
a_\nu^{-1}=&\omega_q(\rho_\nu-1/\rho_\nu)\prod_{j\neq\nu}(\rho_\nu+1/\rho_\nu-\rho_j-1/\rho_j)\\
=&-\omega_q\left\{2\psi_\nu\lambda^{-1/(2m)}+O(\lambda^{-3/(2m)})\right\}(-1)^{m+1}m\psi_\nu^{-2}\lambda^{-1+1/m}\left\{1+O(\lambda^{-1/m})\right\}\\
=&2m(-1)^m\lambda^{-1+1/(2m)}\omega_q\psi_\nu^{-1}\left\{1+O(\lambda^{-1/m})\right\}\\
=&2m\lambda^{1/(2m)}\psi_\nu^{-1}\left\{1+O(\lambda^{-1/m})\right\}.
\end{aligned}$$

The above derivation establishes (A.26).

The case $p>m$

To derive a_ν, we need to study (A.28) again. For the term $\rho_\nu+\rho_\nu^{-1}-\rho_j-\rho_j^{-1}$ in (A.28), there are two new cases besides (A.29),

$$\rho_\nu+\rho_\nu^{-1}-\rho_j-\rho_j^{-1}=\begin{cases}-\psi_j^{-1}(\lambda/\omega_q)^{1/(p-m)}+O(1), & \nu\leq m<j\,,\\ (\lambda/\omega_q)^{1/(p-m)}(\psi_\nu^{-1}-\psi_j^{-1})+O(1), & \nu>m,j>m\,.\end{cases}$$

It is easy to show when $\nu>m$, a_ν is of order $\lambda^{p/(m-p)}$ and when $1\leq\nu\leq m$, (A.26) is still valid. Notice that in this case ω_q is a constant that only depends on p. So now we have finished the proof of Proposition A.4.

A.4.3 Derivation of $\tilde{a}_{k,\nu}$

In this subsection, we shall derive the form of $\tilde{a}_{k,\nu}$ satisfying the constraints in (A.16). Instead of giving a proposition, we derive the form of $\tilde{a}_{k,\nu}$ in the context.

Consider the k's satisfying $k \in (Kx - p - 1, Kx + p + 1)$. Since x goes to 0 at a rate of $\lambda^{1/(2m)}/K$, $k > (p + m)$. Hence $\{\mathbf{S}_k + \mathbf{R}_k(x)\}^T \mathbf{\Lambda}_k = 1$ is automatically satisfied for arbitrary $\tilde{\mathbf{a}}_k$. Denote $\mathbf{P} = \mathbf{D}^T\mathbf{D}$ and $\mathbf{P}_k$ the kth column of $\mathbf{P}$. Note that every row of $\mathbf{B}^T\mathbf{B}/M$ sums to 1, hence

$$\{\mathbf{S}_k + \mathbf{R}_k(x)\}^T (\mathbf{\Lambda}_j - \lambda \mathbf{P}_j) = O\{\lambda^{-1/(2m)}\} + O\left(\max_{1\le\nu\le q} |\tilde{a}_{k,\nu}|\right), \quad j = 1, \ldots, q.$$

In light of the constraints in (A.16),

$$\{\mathbf{S}_k + \mathbf{R}_k(x)\}^T \mathbf{P}_j = O\left\{\lambda^{-1-1/(2m)}\right\} + \lambda^{-1} O\left(\max_{1\le\nu\le q} |\tilde{a}_{k,\nu}|\right), \quad j = 1, \ldots, q.$$

For simplicity, denote $O\left\{\lambda^{-1-1/(2m)}\right\} + \lambda^{-1} O\left(\max_{1\le\nu\le q} |\tilde{a}_{k,\nu}|\right)$ by ξ. Further simplification shows that the above is equivalent to

$$\sum_{\nu=1}^{q} (1 - \rho_\nu^{-1})^{m+j-1} a_\nu \rho_\nu^{k-1} + \sum_{\nu=1}^{q} (1 - \rho_\nu)^{m+j-1} \tilde{a}_{k,\nu} = O(\xi), \quad j = 1, \ldots, m, \quad \text{(A.32)}$$

and if $p > m$,

$$\sum_{\nu=1}^{q} (1-\rho_\nu^{-1})^{2m} \rho_\nu^{-(j-m-1)} a_\nu \rho_\nu^{k-1} + \sum_{\nu=1}^{q} (1-\rho_\nu)^{2m} \rho_\nu^{j-m-1} \tilde{a}_{k,\nu} = O(\xi), \quad j = m+1, \ldots, q. \quad \text{(A.33)}$$

The case $p \le m$

Because $k \in (Kx - p - 1, Kx + p + 1)$, $k/\{c_x \lambda^{1/(2m)}\} \to 1$. Hence for $1 \le \nu \le m$, $\rho_\nu^{k-1} \to \exp(-c_x \psi_\nu)$. Since $q = m$, all ρ_ν's take the forms in (A.19). As $\lambda \to \infty$, $\rho_\nu \to 1$, $(1 - \rho_\nu)^j \to \psi_\nu^j \lambda^{-j/(2m)}$, $(1 - \rho_\nu^{-1})^j \to (-1)^j \psi_\nu^j \lambda^{-j/(2m)}$ and $a_\nu \to$

$\frac{1}{2m}\psi_\nu\lambda^{-1/(2m)}$. It is easy to show the leading term of $\sum_{\nu=1}^m(1-\rho_\nu^{-1})^{m+j-1}a_\nu\rho_\nu^{k-1}$ is $(2m)^{-1}\lambda^{-(m+j)/(2m)}\sum_{\nu=1}^m(-1)^{m+j-1}\psi_\nu^{m+j}\exp(-c_x\psi_\nu)$ and the leading term of $\sum_{\nu=1}^m(1-\rho_\nu)^{m+j-1}\tilde{a}_{k,\nu}$ is $\lambda^{-(m+j-1)/(2m)}\sum_{\nu=1}^m\psi_\nu^{m+j-1}\tilde{a}_{k,\nu}$. Therefore, we derive that

$$\tilde{a}_{k,\nu}=\frac{\tilde{b}_{k,\nu}}{2m}\lambda^{-1/(2m)}+O(\lambda^{-1/m}),\quad 1\le\nu\le m, \tag{A.34}$$

for some constant $\tilde{b}_{k,\nu}$. Because of (A.34), $\xi=O\{\lambda^{-1-1/(2m)}\}$. Matching the coefficients of $\lambda^{-(m+j)/(2m)}$ for the jth term in (A.32) gives

$$\sum_{\nu=1}^m(-1)^{m+j-1}\psi_\nu^{m+j}\exp(-c_x\psi_\nu)+\sum_{\nu=1}^m\psi_\nu^{m+j-1}\tilde{b}_{k,\nu}=0 \tag{A.35}$$

To simplify notation, we define $\mathbf{\Psi}_{m,1}$ is an $m\times m$ matrix with its (i,j)th element ψ_j^{m+i-1}, $\mathbf{\Psi}_{m,2}$ is an $m\times m$ matrix with its (i,j)th element $(-1)^{m+j}\psi_j^{m+i}$ and $\mathbf{r}(x)=(e^{-\psi_1 x},\ldots,e^{-\psi_m x})^T$. By (A.35),

$$(\tilde{b}_{k,1},\ldots,\tilde{b}_{k,m})^T=\mathbf{\Psi}_{m,1}^{-1}\mathbf{\Psi}_{m,2}\mathbf{r}(c_x). \tag{A.36}$$

The case $p>m$

Note that if $\nu>m$, $\rho_\nu=O\{\lambda^{-1/(p-m)}\}$ and $a_\nu=O\{\lambda^{-p/(p-m)}\}$. Equality (A.33) for $j=m+1$ reduces to

$$(-1)^{m+1}\lambda^{-1-1/(2m)}\sum_{\nu=1}^m\psi_\nu\exp(-c_x\psi_\nu)+(-1)^{m+1}\lambda^{-1}\sum_{\nu=1}^m\tilde{a}_{k,\nu}+\sum_{\nu=m+1}^q\tilde{a}_{k,\nu}=O(\xi),$$

i.e.,

$$\sum_{\nu=m+1}^q\tilde{a}_{k,\nu}=\lambda^{-1}(-1)^{m+1}\sum_{\nu=1}^m\tilde{a}_{k,\nu}+O(\xi)=O(\xi). \tag{A.37}$$

Because of (A.37), the analysis in the previous subsection is also valid and (A.36) still holds. Furthermore, we can derive from (A.33) that

$$\sum_{\nu=m+1}^q\tilde{a}_{k,\nu}\rho_\nu^j=O\left\{\lambda^{-1-1/(2m)}\right\},\quad j=0,\ldots,q-m-1. \tag{A.38}$$

It follows from (A.38) that

$$\sum_{\nu=m+1}^{q} \tilde{a}_{k,\nu}\rho_\nu^j = O\left\{\lambda^{-1-1/(2m)}\right\}, \text{ for any non-negative integer } j. \tag{A.39}$$

A.5 Derivation of Asymptotics

In this section, we shall prove the main results in Section A.3. Specifically, we shall derive the asymptotic distribution of P-splines when $x \in (0,1)$ and when x goes to 0 at certain rate. Define $\bar{x}_k = (k-1/2)/K$.

A.5.1 The Case $x \in (0,1)$

To prove Proposition A.1, we need Proposition A.5 below.

Proposition A.5 *Let* $h_n = \lambda^{1/(2m)}/K$. *Let* $\psi_0 = \min\{Re(\psi_1), \ldots, Re(\psi_m)\}$, *where* $Re(\cdot)$ *gives the real part of a complex number. Assume* $h_n = o(1)$ *and* $(Kh_n)^{-1} = o(1)$. *For* $x \in (0,1)$,

$$\begin{aligned}
&nh_n \sum_{k,r} B_k(x)B_r(x_i)S_{k,r}/M \\
=&H_m\left(\frac{|x-x_i|}{h_n}\right) + \delta_{\{p>m\}}\left[O\left(\lambda^{-2+\frac{1}{2m}}\right) + \delta_{\{|x-x_i|<(3p+2-m)/K\}}O\left(\lambda^{-\frac{p}{p-m}+\frac{1}{2m}}\right)\right] \\
&+\exp\left(-\psi_0\frac{|x-x_i|}{h_n}\right)\left[O\left(\lambda^{-1/m}\right) + \delta_{\{m=1\}}\delta_{\left\{|x-x_i|\le(p+1)\lambda^{-1/(2m)}\right\}}O\left\{\lambda^{-1/(2m)}\right\}\right].
\end{aligned} \tag{A.40}$$

Here $\delta_{\{p>m\}} = 1$ *if* $p > m$ *and 0 otherwise; the other* δ *terms are similarly defined.*

Proof of Proposition A.5: By the definition of $\mathbf{S}_k$ in (A.11),

$$\sum_{k,r} B_k(x)B_r(x_i)S_{k,r}/M = \sum_{\nu=1}^{q}\left\{\sum_{k,r} B_k(x)B_r(x_i)a_\nu\rho_\nu^{|k-r|}/M\right\}.$$

If $p > m$ and $\nu > m$, $\rho_\nu = O\{\lambda^{-1/(p-m)}\}$ by Proposition A.3 and a_ν is of order $\lambda^{-p/(p-m)}$ by Proposition A.4. Note that if $|x - x_i| \geq (3p+2-m)/K$, a necessary condition for a nonzero $B_k(x)B_r(x_i)$ is that $|k-r| \geq p-m$, hence, for $\nu > m$,

$$\begin{aligned}&\sum_{k,r} B_k(x)B_r(x_i)a_\nu\rho_\nu^{|k-r|}/M \\ =&\delta_{\{|x-x_i|<(3p+2-m)/K\}}O\left\{\lambda^{-p/(p-m)}Kn^{-1}\right\} + O(\lambda^{-2}Kn^{-1}).\end{aligned} \tag{A.41}$$

In the above derivation, Lemma A.2 was used. Fix $1 \leq \nu \leq m$. Define

$$b_\nu = -\lambda^{1/(2m)}\log(\rho_\nu), \quad 1 \leq \nu \leq m.$$

Then by (A.19),

$$b_\nu = \psi_\nu + O\left(\lambda^{-1/m}\right), \quad 1 \leq \nu \leq m.$$

It follows that

$$\rho_\nu^{|k-r|} = \exp\left(-b_\nu\frac{|\bar{x}_k - \bar{x}_r|}{h_n}\right) = \exp\left(-\psi_\nu\frac{|\bar{x}_k - \bar{x}_r|}{h_n}\right)\left\{1 + \frac{|\bar{x}_k - \bar{x}_r|}{h_n}O\left(\lambda^{-1/m}\right)\right\}.$$

By the expression of a_ν in (A.26),

$$a_\nu\rho_\nu^{|k-r|} = \frac{\psi_\nu}{2mKh_n}\exp\left(-\psi_\nu\frac{|\bar{x}_k - \bar{x}_r|}{h_n}\right)\left\{1 + \left(1 + \frac{|\bar{x}_k - \bar{x}_r|}{h_n}\right)O\left(\lambda^{-1/m}\right)\right\}.$$

In light of Lemma A.7,

$$\begin{aligned}&2mnh_n\left\{\sum_{k,r} B_k(x)B_r(x_i)a_\nu\rho_\nu^{|k-r|}/M\right\} \\ =&\sum_{k,r} B_k(x)B_r(x_i)\psi_\nu\exp\left(-\psi_\nu\frac{|\bar{x}_k - \bar{x}_r|}{h_n}\right)\left\{1 + \left(1 + \frac{|\bar{x}_k - \bar{x}_r|}{h_n}\right)O\left(\lambda^{-1/m}\right)\right\} \\ =&\psi_\nu\exp\left(-\psi_\nu\frac{|x - x_i|}{h_n}\right)\left\{1 - \frac{\psi_\nu}{Kh_n}\tilde{g}(x, x_i) + O\left(\lambda^{-1/m}\right)\right\}.\end{aligned} \tag{A.42}$$

Summing (A.42) for $\nu = 1, \ldots, m$ gives

$$\begin{aligned}
&nh_n \left\{ \sum_{\nu=1}^{m} \sum_{k,r} B_k(x) B_r(x_i) a_\nu \rho_\nu^{|k-r|} / M \right\} \\
=&\frac{1}{2m} \sum_{\nu=1}^{m} \psi_\nu \exp\left(-\psi_\nu \frac{|\bar{x} - x_i|}{h_n} \right) \left\{ 1 - \frac{\psi_\nu}{K h_n} \tilde{g}(x, x_i) + O\left(\lambda^{-1/m}\right) \right\} \\
=&H_m \left(\frac{|x - x_i|}{h_n} \right) + \exp\left(-\psi_0 \frac{|x - x_i|}{h_n} \right) O\left(\lambda^{-1/m}\right) - \frac{1}{K h_n} \tilde{g}(x, x_i) Q\left(\frac{|x - x_i|}{h_n} \right),
\end{aligned} \tag{A.43}$$

where

$$Q(x) = \frac{1}{2m} \sum_{\nu=1}^{m} \psi_\nu^2 \exp\left(-\psi_\nu |x|\right).$$

It is easy to show that $|Q(x)| \leq \exp(-\psi_0 |x|)$. Lemma A.8 states that $\tilde{g}(x, x_i) = 0$ if $|x - x_i| \geq (p+1)/K$. Lemma A.12 states when $m > 1$, $\sum_{1 \leq \nu \leq m} \psi_\nu^2 = 0$. Thus if x is close to 0 and $m > 1$, $\sum_{1 \leq \nu \leq m} \psi_\nu^2 \exp(-\psi_\nu |x|)$ is of the same order as x. Hence,

$$\begin{aligned}
&\tilde{g}(x, x_i) Q\left(\frac{|x - x_i|}{h_n} \right) \\
=&\delta_{\{|x - x_i| \leq (p+1)/(K h_n)\}} \exp\left(-\psi_0 \frac{|x - x_i|}{h_n} \right) \left[O\left\{ (K h_n)^{-2} \right\} + \delta_{\{m=1\}} O\left\{ (K h_n)^{-1} \right\} \right].
\end{aligned} \tag{A.44}$$

Equalities (A.41)–(A.44) together prove Proposition A.5.

Proof of Proposition A.1: By (A.15) and Proposition A.5,

$$\hat{\mu}(x) = \frac{1}{n h_n} \sum_{i=1}^{n} y_i \left\{ H_m \left(\frac{|x - x_i|}{h_n} \right) + r_i(x) \right\} = \mu^*(x) + \frac{1}{n h_n} \sum_{i=1}^{n} r_i(x) y_i,$$

where

$$\begin{aligned}
r_i(x) =& \exp\left(-\psi_0 \frac{|x - x_i|}{h_n} \right) \left[O\left(\lambda^{-\frac{1}{m}} \right) + \delta_{\{m=1\}} \delta_{\left\{ |x - x_i| \leq (p+1)\lambda^{-1/(2m)} \right\}} O\left(\lambda^{-\frac{1}{2m}} \right) \right] \\
&+ \delta_{(p>m)} \left[O\left(\lambda^{-2+\frac{1}{2m}} \right) + \delta_{\{|x - x_i| < (3p+2-m)/K\}} O\left\{ \lambda^{-\frac{p}{p-m} + \frac{1}{2m}} \right\} \right] \\
&+ O\left[n h_n \exp\{ -C \lambda^{-\frac{1}{2m}} K \min(x, 1-x) \} \right].
\end{aligned} \tag{A.45}$$

First we have

$$|\mathrm{E}\{\hat{\mu}(x) - \mu^*(x)\}| \leq (nh_n)^{-1} \sum_i |\mu(x_i) r_i(x)| . \tag{A.46}$$

We study the right hand side of (A.46). For $r_i(x)$ defined in (A.45), the two terms $O\{\lambda^{-2+1/(2m)}\}$ and $O\left[nh_n \exp\{-C\lambda^{-1/(2m)} K \min(x, 1-x)\}\right]$ are of order $o(\lambda^{-1/m})$. Also

$$(nh_n)^{-1} \sum_i |\mu(x_i)| \exp\left(-\psi_0 \frac{|x - x_i|}{h_n}\right) = O(1),$$

$$(nh_n)^{-1} \sum_i |\mu(x_i)| \exp\left(-\psi_0 \frac{|x - x_i|}{h_n}\right) \delta_{\{|x-x_i| \leq (p+1)\lambda^{-1/(2m)}\}} = O\left(\lambda^{-\frac{1}{2m}}\right),$$

$$(nh_n)^{-1} \sum_i |\mu(x_i)| \delta_{\{|x-x_i| \leq (3p+2-m)/K\}} = O\{(Kh_n)^{-1}\}.$$

It follows that $\sum_i |\mu(x_i) r_i(x)| = O(\lambda^{-1/m})$. Next we derive that

$$\mathrm{var}\{\hat{\mu}(x) - \mu^*(x)\} = (nh_n)^{-2} \sum_i r_i^2(x) \sigma^2(x_i). \tag{A.47}$$

With similar derivation as before, we can establish that $(nh_n)^{-1} \sum_i r_i^2(x)\sigma^2(x_i) = o(1)$. Therefore the proposition is proved.

Example A.1 *Consider the case $m = 2$. Denote the imaginary number by $\imath$. Then $\psi_1 = \frac{1+\imath}{\sqrt{2}}$ and $\psi_2 = \frac{1-\imath}{\sqrt{2}}$. Hence the equivalent kernel for $x \in (0, 1)$ is*

$$\frac{1}{2\sqrt{2}} e^{-\frac{|x-\tilde{x}|}{\sqrt{2}}} \left\{\cos\left(\frac{|x-\tilde{x}|}{\sqrt{2}}\right) + \sin\left(\frac{|x-\tilde{x}|}{\sqrt{2}}\right)\right\}.$$

Example A.2 *Consider the case $m = 3$. Then $\psi_1 = 1, \psi_2 = \frac{1+\sqrt{3}\imath}{2}, \psi_3 = \frac{1-\sqrt{3}\imath}{2}$. Hence the equivalent kernel for $x \in (0, 1)$ is*

$$\frac{1}{6} e^{-|x-\tilde{x}|} + \frac{1}{6} e^{-\frac{|x-\tilde{x}|}{2}} \left\{\cos\left(\frac{\sqrt{3}|x-\tilde{x}|}{2}\right) + \sqrt{3} \sin\left(\frac{\sqrt{3}|x-\tilde{x}|}{2}\right)\right\}.$$

Proof of Theorem A.1: Proposition A.1 shows that the P-spline estimator is asymptotically equivalent to a kernel regression estimator with the kernel function

$H_m(x)$. Hence a standard analysis of the kernel regression estimator as in Wand and Jones (1995) with the kernel function $H_m(x)$ should give us the desired result. The detailed derivation is as follows. First,

$$\mathrm{E}\{\mu^*(x)\} = \mu(x) + (-1)^{m+1} h_n^{2m} \mu^{(2m)}(x) + o(h_n^{2m})$$

and

$$\begin{aligned}\mathrm{var}\,\{\mu^*(x)\} &= \sum_i \sigma^2(x_i) \frac{1}{(nh_n)^2} H_m^2\left(\frac{|x - x_i|}{h_n}\right) \\ &= \frac{1}{nh_n}\sigma^2(x) \int_{-\infty}^{\infty} H_m^2(s)ds + o\{(nh_n)^{-1}\}.\end{aligned}$$

By Proposition A.1, we obtain

$$\begin{aligned}\mathrm{E}\{\hat{\mu}(x)\} &= \mu(x) + (-1)^{m+1} h_n^{2m} \mu^{(2m)}(x) + o(h_n^{2m}) + O\{(nh_n)^{-1}\}, \\ \mathrm{var}\,\{\hat{\mu}(x)\} &= \frac{1}{nh_n}\sigma^2(x) \int_{-\infty}^{\infty} H_m^2(s)ds + o\{(nh_n)^{-1}\},\end{aligned}$$

and the proof is straightforward by verifying that h_n^{4m} and $(nh_n)^{-1}$ are of the same order and $\lambda^{-1/m} = o(h_n^{2m})$.

A.5.2 The Boundary Case

By (A.17) and the derivation in Section A.4.3, we have

$$\begin{aligned}\hat{\mu}(x) &= \frac{1}{M}\sum_{i=1}^{n} y_i \left[\sum_{k,r} B_k(x) B_r(x_i) \{S_{k,r} + R_{k,r}(x)\} + b_{i,0}(x)\right] \\ &= \frac{1}{M}\sum_{i=1}^{n} y_i \left\{\sum_{k,r} B_k(x) B_r(x_i) S_{k,r} + b_{i,0}(x)\right\} \qquad \text{(A.48)} \\ &\quad + \frac{1}{M}\sum_{i=1}^{n} y_i \left\{\sum_{k,r} B_k(x) B_r(x_i) R_{k,r}(x)\right\}. \qquad \text{(A.49)}\end{aligned}$$

Note that $b_{i,0}(x) = O[\exp\{-C_0\lambda^{-1/(2m)}K\}]$. The sum in (A.48) can be similarly analyzed as in Section A.5.1 and we have

$$\begin{aligned}&\frac{1}{M}\sum_{i=1}^{n} y_i \left\{\sum_{k,r} B_k(x)B_r(x_i)S_{k,r} + b_{i,0}(x)\right\}\\ =&\frac{1}{nh_n}\sum_{i=1}^{n} y_i \left[H_m\left(\frac{|x-x_i|}{h_n}\right) + \exp\left(-\psi_0\frac{|x-x_i|}{h_n}\right) O\left\{(Kh_n)^{-1}\right\}\right]\end{aligned}$$

Now we focus on the second sum (denoted by $\hat{\mu}_b(x)$) in (A.49). Note that $R_{k,r}(x) = \sum_{\nu=1}^{q} \tilde{a}_{k,\nu}\rho_\nu^{r-1}$. Note also if $\nu > m$, $\rho_\nu = O\{\lambda^{-1/(p-m)}\}$ and (A.39) holds. Hence,

$$\hat{\mu}_b(x) = \frac{1}{2mnh_n}\sum_{i=1}^{n} y_i \left[\sum_{\nu=1}^{m}\sum_{r=1}^{c}\sum_{k=1}^{c} B_r(x_i)B_k(x)\tilde{b}_{k,\nu}\rho_\nu^{r-1} + O\{(Kh_n)^{-2}\}\right].$$

By a similar analysis as in Section A.5.1, we obtain, aided by Lemma A.9, that

$$\begin{aligned}\hat{\mu}_b(x) &= \frac{1}{2mnh_n}\sum_{i=1}^{n} y_i \left[\mathbf{r}^T(\frac{x_i}{h_n})\mathbf{\Psi}_{m,1}^{-1}\mathbf{\Psi}_{m,2}\mathbf{r}(c_x) + O\{(Kh_n)^{-2}\}\right]\\ &= \frac{1}{2mnh_n}\sum_{i=1}^{n} y_i \left[\mathbf{r}^T(\frac{x_i}{h_n})\mathbf{\Psi}_{m,1}^{-1}\mathbf{\Psi}_{m,2}\mathbf{r}(\frac{x}{h_n}) + O\{(Kh_n)^{-2}\}\right].\end{aligned}$$

Note that $\mathbf{\Psi}_{m,1}$, $\mathbf{\Psi}_{m,2}$ and $\mathbf{r}(x)$ are defined in Section A.4.3. In the above derivation, we used the assumption that x/h_n converges to c_x; we also used (A.36). We define the equivalent kernel for $\hat{\mu}_b(x)$ as

$$H_{b,m}(x,\tilde{x}) = \frac{1}{2m}\mathbf{r}(\tilde{x})^T\mathbf{\Psi}_{m,1}^{-1}\mathbf{\Psi}_{m,2}\mathbf{r}(x). \tag{A.50}$$

Now we have

$$\hat{\mu}(x) = \frac{1}{nh_n}\sum_{i=1}^{n} y_i \left[H_m\left(\frac{|x-x_i|}{h_n}\right) + H_{b,m}\left(\frac{x}{h_n},\frac{x_i}{h_n}\right) + \exp\left(-\psi_0\frac{x_i}{h_n}\right) O\left(\frac{1}{Kh_n}\right)\right]. \tag{A.51}$$

The above equality shows that when x is near 0, a P-spline estimator is a kernel regression estimator with the equivalent kernel

$$H_m(|x-\tilde{x}|) + H_{b,m}(x,\tilde{x}). \tag{A.52}$$

Next we provide two specific examples of (A.52).

Example A.3 *Consider the case $m = 2$. It can be shown that*

$$\Psi_{m,1} = \begin{pmatrix} \imath & -\imath \\ \\ \frac{-1+\imath}{\sqrt{2}} & \frac{-1-\imath}{\sqrt{2}} \end{pmatrix}, \quad \Psi_{m,2} = \begin{pmatrix} -\frac{-1+\imath}{\sqrt{2}} & -\frac{-1-\imath}{\sqrt{2}} \\ \\ -1 & -1 \end{pmatrix},$$

and

$$\mathbf{r}(x) = e^{-\frac{x}{\sqrt{2}}} \begin{pmatrix} cos\left(\frac{x}{\sqrt{2}}\right) - \imath \sin\left(\frac{x}{\sqrt{2}}\right) \\ \\ \cos\left(\frac{x}{\sqrt{2}}\right) + \imath \sin\left(\frac{x}{\sqrt{2}}\right) \end{pmatrix}.$$

Hence,

$$H_{b,2}(x, \tilde{x}) = \frac{\sqrt{2}}{4} e^{-\frac{|x+\tilde{x}|}{\sqrt{2}}} \left\{ \cos\left(\frac{|x-\tilde{x}|}{\sqrt{2}}\right) + 2\cos\left(\frac{x}{\sqrt{2}}\right)\cos\left(\frac{\tilde{x}}{\sqrt{2}}\right) - \sin\left(\frac{x+\tilde{x}}{\sqrt{2}}\right) \right\}.$$

It follows that the equivalent kernel for x near 0 is

$$\begin{aligned} &\frac{\sqrt{2}}{4} e^{-\frac{|x-\tilde{x}|}{\sqrt{2}}} \left\{ \cos\left(\frac{|x-\tilde{x}|}{\sqrt{2}}\right) + \sin\left(\frac{|x-\tilde{x}|}{\sqrt{2}}\right) \right\} \\ &+ \frac{\sqrt{2}}{4} e^{-\frac{|x+\tilde{x}|}{\sqrt{2}}} \left\{ \cos\left(\frac{|x-\tilde{x}|}{\sqrt{2}}\right) + 2\cos\left(\frac{x}{\sqrt{2}}\right)\cos\left(\frac{\tilde{x}}{\sqrt{2}}\right) - \sin\left(\frac{x+\tilde{x}}{\sqrt{2}}\right) \right\}. \end{aligned}$$

When $x = 0$, the equivalent kernel becomes

$$\sqrt{2} e^{-\tilde{x}/\sqrt{2}} \cos\left(\tilde{x}/\sqrt{2}\right),$$

which coincides with the equivalent kernel for the smoothing splines (Silverman, 1984).

Example A.4 *Consider the case $m = 3$. It can be shown that*

$$\Psi_{m,1} = \begin{pmatrix} 1 & -1 & -1 \\ \\ 1 & \frac{-1-\sqrt{3}\imath}{2} & \frac{-1+\sqrt{3}\imath}{2} \\ \\ 1 & \frac{1-\sqrt{3}\imath}{2} & \frac{1+\sqrt{3}\imath}{2} \end{pmatrix}, \quad \Psi_{m,2} = \begin{pmatrix} 1 & \frac{-1-\sqrt{3}\imath}{2} & \frac{-1+\sqrt{3}\imath}{2} \\ \\ 1 & \frac{1-\sqrt{3}\imath}{2} & \frac{1+\sqrt{3}\imath}{2} \\ \\ 1 & 1 & 1 \end{pmatrix},$$

and

$$\mathbf{r}(x) = \begin{pmatrix} e^{-x} \\ e^{-\frac{x}{2}} \left\{ cos\left(\frac{\sqrt{3}x}{2}\right) - \imath \sin\left(\frac{\sqrt{3}x}{2}\right) \right\} \\ e^{-\frac{x}{2}} \left\{ \cos\left(\frac{\sqrt{3}x}{2}\right) + \imath \sin\left(\frac{\sqrt{3}x}{2}\right) \right\} \end{pmatrix}.$$

It follows that the equivalent kernel for x near 0 is

$$\begin{aligned}
&\frac{1}{6}e^{-|x-\tilde{x}|} + \frac{1}{6}e^{-\frac{|x-\tilde{x}|}{2}} \left\{ \cos\left(\frac{\sqrt{3}|x-\tilde{x}|}{2}\right) + \sqrt{3}\sin\left(\frac{\sqrt{3}|x-\tilde{x}|}{2}\right) \right\} \\
&+\frac{3}{6}e^{-|x+\tilde{x}|} + \frac{2}{6}e^{-|x+\frac{\tilde{x}}{2}|} \left\{ \cos\left(\frac{\sqrt{3}\tilde{x}}{2}\right) - \sqrt{3}\sin\left(\frac{\sqrt{3}\tilde{x}}{2}\right) \right\} \\
&+\frac{2}{6}e^{-|\tilde{x}+\frac{x}{2}|} \left\{ \cos\left(\frac{\sqrt{3}x}{2}\right) - \sqrt{3}\sin\left(\frac{\sqrt{3}x}{2}\right) \right\} \\
&+\frac{1}{6}e^{-\frac{|x+\tilde{x}|}{2}} \left\{ 3\cos\left(\frac{\sqrt{3}(\tilde{x}-x)}{2}\right) - \sqrt{3}\sin\left(\frac{\sqrt{3}(\tilde{x}-x)}{2}\right) + 2\sin\left(\frac{\sqrt{3}x}{2}\right)\sin\left(\frac{\sqrt{3}\tilde{x}}{2}\right) \right\}.
\end{aligned}$$

When $x = 0$, the equivalent kernel becomes

$$e^{-\tilde{x}} + e^{-\tilde{x}/2} \left\{ \cos\left(\frac{\sqrt{3}\tilde{x}}{2}\right) - \frac{\sqrt{3}}{3}\sin\left(\frac{\sqrt{3}\tilde{x}}{2}\right) \right\}.$$

Proof of Theorem A.2: Similar to the proof of Theorem A.1, we can derive that

$$\begin{aligned}
&\mathrm{E}\{\hat{\mu}(x)\} \\
=&\frac{1}{nh_n}\sum_{i=1}^{n}\mu(x_i)\left[H_m\left(\frac{|x-x_i|}{h_n}\right) + H_{b,m}\left(\frac{x}{h_n},\frac{x_i}{h_n}\right) + \exp\left(-\psi_0\frac{x_i}{h_n}\right)O\left(\frac{1}{Kh_n}\right)\right] \\
=&\frac{1}{h_n}\int_0^1 \mu(u)\left\{H_m\left(\frac{|x-u|}{h_n}\right) + H_{b,m}\left(\frac{x}{h_n},\frac{u}{h_n}\right)\right\}du + O\left(\frac{1}{Kh_n}\right) \\
=&\int_{-\infty}^{c_x}\mu(x-hv)\left\{H_m(v) + H_{b,m}(c_x, c_x-v)\right\}dv + O\left\{(Kh_n)^{-1}\right\},
\end{aligned}$$

and

$$
\begin{aligned}
&\operatorname{var}\{\hat{\mu}(x)\} \\
=&\frac{1}{(nh_n)^2}\sum_{i=1}^{n}\sigma^2(x_i)\left[H_m\left(\frac{|x-x_i|}{h_n}\right)+H_{b,m}\left(\frac{x}{h_n},\frac{x_i}{h_n}\right)+\exp\left(-\psi_0\frac{x_i}{h_n}\right)O\left(\frac{1}{Kh_n}\right)\right]^2 \\
=&\frac{1+o(1)}{nh_n}\frac{1}{h_n}\int_0^1\sigma^2(u)\left\{H_m\left(\frac{|x-u|}{h_n}\right)+H_{b,m}\left(\frac{x}{h_n},\frac{u}{h_n}\right)\right\}^2du \\
=&\frac{1+o(1)}{nh_n}\sigma^2(x)\int_{-\infty}^{c_x}\left\{H_m(v)+H_{b,m}(c_x,c_x-v)\right\}^2dv.
\end{aligned}
\tag{A.53}
$$

By Proposition A.6 below, we have

$$
\begin{aligned}
\mathrm{E}\{\hat{\mu}(x)\}=&\mu(x)+(-1)^{m+1}h_n^m\mu^{(m)}(x)\int_{-\infty}^{c_x}v^m\left\{H_m(v)+H_{b,m}(c_x,c_x-v)\right\}dv \\
&+o\left(h_n^{m+1}\right)+O\left\{(Kh_n)^{-1}\right\}.
\end{aligned}
\tag{A.54}
$$

Combining (A.53) with (A.54), Theorem A.2 is proved.

Proposition A.6 *For any fixed constant $t \geq 0$,*

$$
\int_{-\infty}^{t}x^{\ell}\left\{H_m(x)+H_{b,m}(t,t-x)\right\}dx=0,\quad \ell=1,\ldots,m-1,
$$

and

$$
\int_{-\infty}^{t}x^{m}\left\{H_m(x)+H_{b,m}(t,t-x)\right\}dx\neq 0.
$$

Proof of Proposition A.6: By Lemma A.10, we can show that

$$
\begin{aligned}
\int_{-\infty}^{t}x^{\ell}H_m(x)\mathrm{d}x&=-\frac{\ell!}{2m}\sum_{k=1}^{\ell+1}\sum_{\nu=1}^{m}\frac{t^{\ell-k+1}}{(\ell-k+1)|}\bar{\psi}_{\nu}^{k-1}e^{-\psi_\nu t} \\
&=-\frac{\ell!}{2m}\left\{\sum_{k=1}^{\ell+1}\frac{t^{\ell-k+1}}{(\ell-k+1)!}\bar{\psi}_1^{k-1},\ldots,\sum_{k=1}^{\ell+1}\frac{t^{\ell-k+1}}{(\ell-k+1)!}\bar{\psi}_m^{k-1}\right\}\mathbf{r}(t),
\end{aligned}
$$

and

$$
\int_{-\infty}^{t}x^{\ell}\mathbf{r}(t-x)^T\mathrm{d}x=-\ell!\left\{\sum_{k=1}^{\ell+1}\frac{t^{\ell-k+1}}{(\ell-k+1)!}(-1)^k\bar{\psi}_1^k,\ldots,\sum_{k=1}^{\ell+1}\frac{t^{\ell-k+1}}{(\ell-k+1)!}(-1)^k\bar{\psi}_m^k\right\}.
$$

Because $H_{b,m}(t, t-x) = (2m)^{-1}\mathbf{r}(t-x)^T\boldsymbol{\Psi}_{m,1}^{-1}\boldsymbol{\Psi}_{m,2}\mathbf{r}(t)$, it suffices to prove that

$$\left(\bar{\psi}_1^{k-1}, \ldots, \bar{\psi}_m^{k-1}\right) + (-1)^k \left(\bar{\psi}_1^{k}, \ldots, \bar{\psi}_m^{k}\right) \boldsymbol{\Psi}_{m,1}^{-1}\boldsymbol{\Psi}_{m,2} = \mathbf{0}^T, \quad k = 1, \ldots, m. \quad \text{(A.55)}$$

Let $\mathbf{w}_k^T = (-1)^{m+1}\left(\bar{\psi}_1^{k}, \ldots, \bar{\psi}_m^{k}\right)\boldsymbol{\Psi}_{m,1}^{-1}\boldsymbol{\Psi}_{m,2}$. Then $\mathbf{w}_k$ is the $(m+1-k)$th row of $\boldsymbol{\Psi}_{m,2}$. Hence, for $k = 1, \ldots, m$,

$$\mathbf{w}_k^T = (-1)^{2m-k+1}\left(\psi_1^{2m-k+1}, \ldots, \psi_m^{2m-k+1}\right) = (-1)^{m-k}\left(\bar{\psi}_1^{k-1}, \ldots, \bar{\psi}_m^{k-1}\right)$$

which proves (A.55). For $\ell = m$, we have

$$\int_{-\infty}^{t} x^m \left\{H_m(x) + H_{b,m}(t, t-x)\right\} dx = \frac{-m!}{2m}\tilde{\mathbf{w}}_{m+1}^T\mathbf{r}(t),$$

where $\tilde{\mathbf{w}}_{m+1}^T = \left(\bar{\psi}_1^{m}, \ldots, \bar{\psi}_m^{m}\right) + (-1)^{m+1}\left(\bar{\psi}_1^{m+1}, \ldots, \bar{\psi}_m^{m+1}\right)\boldsymbol{\Psi}_{m,1}^{-1}\boldsymbol{\Psi}_{m,2}$. Note that $(\psi_1^m, \ldots, \psi_m^m) = (-1)^{m+1}\left(\bar{\psi}_1^{m+1}, \ldots, \bar{\psi}_m^{m+1}\right)$ is the first row of $\boldsymbol{\Psi}_{m,1}$, hence

$$\begin{aligned}\tilde{\mathbf{w}}_{m+1}^T &= \left(\bar{\psi}_1^{m}, \ldots, \bar{\psi}_m^{m}\right) + (-1)^{m+1}\left(\psi_1^m, \ldots, \psi_m^m\right) \\ &= 2(-1)^{m+1}\left(\psi_1^m, \ldots, \psi_m^m\right)\end{aligned}$$

which finishes the proof.

A.6 Irregularly Spaced Data

Suppose the design points $\underline{\mathrm{x}} = \{x_1, \ldots, x_n\}$ are independent and sampled from a distribution $F(x)$ in $[0, 1]$. Suppose $F(x)$ is twice continuously differentiable with derivative $f(x)$ and $f(x)$ is positive over $[0, 1]$. For unequally spaced design points, the asymptotic analysis in Section A.5 does not hold here. Instead of pursuing the challenging task of analyzing the P-splines fitted to irregularly spaced data directly, we first bin the data. So we partition $[0, 1]$ into I intervals with equal lengths, and let $\tilde{y}_k$ be the mean of all y_i such that x_i is in the kth bin. If the kth bin has no data point, we let $\tilde{y}_k$ be 0. Here we assume $I \sim c_I n^{\tau_I}$ for some constants

c_I and $\tau_I < 1$. Assuming $\tilde{y}_k$ is the data point at $\tilde{x}_k$, the center of the kth bin, we apply P-splines to the binned data $(\tilde{y}_k)_{1\le k\le I}$ to get

$$\hat{\boldsymbol{\theta}}^* = \boldsymbol{\Lambda}^{-1}\mathbf{B}^T\tilde{\mathbf{y}}/M.$$

Then the penalized estimate is defined as

$$\hat{\mu}(x) = \sum_{k=1}^{c} \hat{\theta}_k^* B_k(x). \tag{A.56}$$

Note that the practice of binning data in penalized splines also appears in Wang and Shen (2010). The asymptotic distribution of $\hat{\mu}(x)$ in (A.56) can be similarly derived as in Section A.5.

Theorem A.3 *Let $\sigma^2(x) = \mathrm{var}(y|X = x)$. Assume $\tau_I > \max(\tau, 1/2)$ and condition (1)-(4) in Proposition A.1 hold. Furthermore, assume $\sigma^2(x)$ has a continuous second derivative. For $x \in (0, 1)$, with the same notation and assumptions as in Theorem A.1, we have that*

$$n^{2m/(4m+1)}\left\{\hat{\mu}(x) - \mu(x)\right\} \Rightarrow N\left\{\tilde{\mu}(x), V(x)/f(x)\right\}$$

in distribution as $n \to \infty$, where $\tilde{\mu}(x)$ is defined in (A.5) and $V(x)$ is defined in (A.6).

Remark A.5 *The above theorem holds for the fixed design as well and the assumption required for the design points is an analogue to (A.59):* $\sup_k |n_k/(nI^{-1}) - f(\tilde{x}_\kappa)| = o(1)$.

Proof of Theorem A.3: By a similar analysis as in Section A.5 to the binned data $\tilde{\mathbf{y}}$ and with n replaced by I, we obtain

$$\hat{\mu}(x) = \frac{1}{Ih_n}\sum_{k=1}^{I} \tilde{y}_k \left\{ H_m\left(\frac{|x - \tilde{x}_k|}{h_n}\right) + r_k(x)\right\},$$

where

$$
\begin{aligned}
r_k(x) = & \exp\left(-\psi_0 \frac{|x-\tilde{x}_k|}{h_n}\right)\left[O\left(\lambda^{-1/m}\right) + \delta_{\{m=1\}}\delta_{\left\{|x-\tilde{x}_k|\leq (p+1)\lambda^{-1/(2m)}\right\}} O\left\{\lambda^{-1/(2m)}\right\}\right] \\
& + \delta_{(p>m)}\left[O\left(\lambda^{-2+\frac{1}{2m}}\right) + \delta_{\{|x-\tilde{x}_k|<(3p+2-m)/K\}} O\left\{\lambda^{-\frac{p}{p-m}+\frac{1}{2m}}\right\}\right] \\
& + O\left[Ih_n \exp\{-C\lambda^{-1/(2m)} K \min(x, 1-x)\}\right].
\end{aligned}
$$

Then

$$
\mathrm{E}\left\{\hat{\mu}(x)|\underline{\mathrm{x}}\right\} = (Ih_n)^{-1} \sum_k \mathrm{E}\left\{\tilde{y}_k|\underline{\mathrm{x}}\right\} \left\{H_m\left(\frac{x-\tilde{x}_k}{h_n}\right) + r_k(x)\right\}, \tag{A.57}
$$

and

$$
\mathrm{var}\left\{\hat{\mu}(x)|\underline{\mathrm{x}}\right\} = (Ih_n)^{-2} \sum_k \mathrm{var}\left\{\tilde{y}_k|\underline{\mathrm{x}}\right\} \left\{H_m\left(\frac{x-\tilde{x}_k}{h_n}\right) + r_k(x)\right\}^2. \tag{A.58}
$$

For simplicity, we let

$$
G_k = H_m\left\{h_n^{-1}(x-\tilde{x}_k)\right\} + b_k(x).
$$

Let n_k be the number of data points in the kth bin, then

$$
\mathrm{var}\left\{\tilde{y}_k|\underline{\mathrm{x}}\right\} = n_k^{-2} \sum_{i=1}^{n} \sigma^2(x_i)\delta_{\{|x_i-\tilde{x}_k|\leq (2I)^{-1}\}}.
$$

So $\mathrm{var}\left\{\sqrt{n_k}\tilde{y}_k|\underline{\mathrm{x}}\right\}$ is a Nadaraya-Watson kernel regression estimator of the conditional variance function $\sigma^2(x)$ at $\tilde{x}_k$. Similarly, $n_k/(nI^{-1})$ is a kernel density estimator of $f(x)$ at $\tilde{x}_k$. By the uniform convergence theory for kernel density estimators and Nadaraya-Watson kernel regression estimators (see, for instance, Hansen (2008)),

$$
\sup_k \left|n_k/(nI^{-1}) - f(\tilde{x}_\kappa)\right| = O_p\left\{\sqrt{I\ln n/n} + I^{-2}\right\} = o_p(1), \tag{A.59}
$$

and

$$
\sup_k \left|\mathrm{var}\left\{\sqrt{n_k}\tilde{y}_k|\underline{\mathrm{x}}\right\} - \sigma^2(\tilde{x}_k)\right| = O_p\left\{\sqrt{I\ln n/n} + I^{-2}\right\} = o_p(1).
$$

It follows that

$$
\sup_k \left|\frac{n}{I}\mathrm{var}\left\{\tilde{y}_k|\underline{\mathrm{x}}\right\} - \frac{\sigma^2(\tilde{x}_\kappa)}{f(\tilde{x}_\kappa)}\right| = o_p(1). \tag{A.60}
$$

Then by (A.58) and (A.60),

$$\left|\text{var}\left\{\hat{\mu}(x)|\underline{\mathrm{x}}\right\} - \frac{1}{nh_nIh_n}\sum_k \frac{\sigma^2(\tilde{x}_\kappa)}{f(\tilde{x}_\kappa)}G_k^2\right| = \frac{o_p(1)}{nh_nIh_n}\sum_k G_k^2 = o_p\left\{(nh_n)^{-1}\right\},$$

and hence

$$\text{var}\left\{\hat{\mu}(x)|\underline{\mathrm{x}}\right\} = \frac{1}{nh_n}\frac{V(x)}{f(x)} + o_p\left\{(nh_n)^{-1}\right\}. \tag{A.61}$$

where $V(x)$ is defined in (A.6). Because

$$\text{E}\left\{\tilde{y}_k|\underline{\mathrm{x}}\right\} = n_k^{-1}\sum_{i=1}^{n}\mu(x_i)\delta_{\{|x_i-\tilde{x}_\kappa|\le(2I)^{-1}\}},$$

we can derive by (A.59) that

$$\sup_k\left|\text{E}\left\{\tilde{y}_k|\underline{\mathrm{x}}\right\} - \mu(\tilde{x}_\kappa)\right| = O_p(I^{-1}).$$

Hence by (A.57),

$$\left|\text{E}\left\{\hat{\mu}(x)|\underline{\mathrm{x}}\right\} - \frac{1}{Ih_n}\sum_k \mu(\tilde{x}_\kappa)G_k\right| = O_p(I^{-1}),$$

and hence

$$\text{E}\left\{\hat{\mu}(x)|\underline{\mathrm{x}}\right\} = \mu(x) + n^{-(2m)/(4m+1)}\tilde{\mu}(x) + o_p\left\{n^{-(2m)/(4m+1)}\right\}, \tag{A.62}$$

where $\tilde{\mu}(x)$ is defined in (A.5). With (A.61) and (A.62), we can derive that

$$n^{(2m)/(4m+1)}\left[\hat{\mu}(x) - \text{E}\left\{\hat{\mu}(x)|\underline{\mathrm{x}}\right\}\right] \Rightarrow N\left\{0, V(x)/f(x)\right\} \tag{A.63}$$

in distribution and

$$n^{(2m)/(4m+1)}\left[\text{E}\left\{\hat{\mu}(x)|\underline{\mathrm{x}}\right\} - \mu(x)\right] = \tilde{\mu}(x) + o_p(1). \tag{A.64}$$

Equalities (A.63) and (A.64) together prove the theorem.

A.7 An Example

We illustrate the idea of binning data using the LIDAR (light detection and ranging) data. The LIDAR data were analyzed in Holst et al. (1996) and Ruppert et al. (1997). The LIDAR data have 221 data points, and details about the LIDAR data can also be found in Ruppert et al. (2003). We fit the response, *logratio*, as a function of the predictor, *range*. First, we fit the data using cubic P-splines with a penalty of second order, and we use 35 equidistant knots as suggested in Ruppert et al. (2003). Then, we fit the binned data using cubic P-splines with a penalty of second order. The number of bins is 60 and we use 15 equidistant knots. The result is given in Figure A.1. We can see that the two fitted curves are similar, with biggest difference occurring when the predictor, *range*, is around 650.

A.8 Discussion

We have concentrated on the asymptotics of penalized splines estimation. In contrast to smoothing splines, penalized splines allow us to choose the number of knots, the degree of splines and the penalty independently. Our study provides theoretical guidelines on how to choose them. In our setting, the penalty λ plays the role of a smoothing parameter and the optimal order for λ is provided. The number of knots K is not important as long as it exceeds a given bound. The choice of the degree of splines does not affect the asymptotic distribution. Our results indicate that the performance of penalized splines estimation is similar to that of smoothing splines estimation (Silverman, 1984) and a class of kernel estimators (Messer and Goldstein, 1993). Furthermore, penalized splines have a slower convergence rate at the boundary than in the interior.

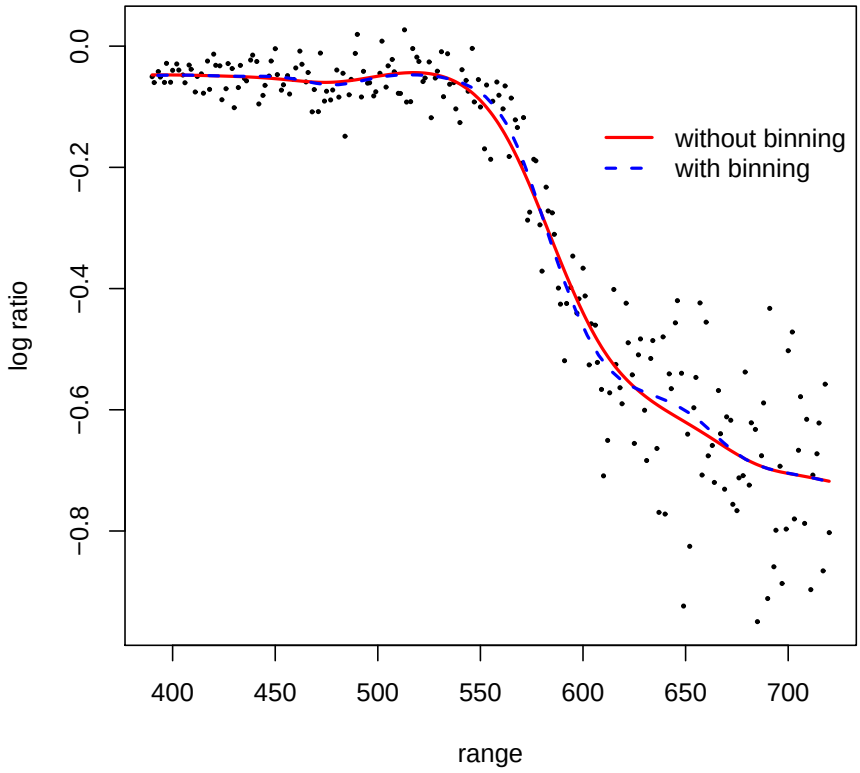


Figure A.1: The fitted curves of the response, log ratio, as a function of the predictor, range. The solid line is the fitted P-splines without binning the data, and the dashed line is the fitted P-splines after binning the data. The solid dots are the observed data.

A.9 Some Lemmas

The following lemmas are needed for this chapter and some of them are also used in Chapter 1.

Lemma A.1 *The coefficients* $\hat{\boldsymbol{\theta}}$ *defined in (A.2) satisfies* $\hat{\theta}_k = \sum_i d_{i,k} y_i$ *with* $d_{i,k} = o(1)$, $1 \leq k \leq c$.

Proof of Lemma A.1: It suffices to show every element of the matrix $(\mathbf{B}^T\mathbf{B} + \lambda^*\mathbf{D}^T\mathbf{D})^{-1}\mathbf{B}^T$ is $o(1)$. Because every column of $\mathbf{B}^T$ contains at most $p+1$ non-zero elements that sum to 1 by Lemma A.2, it suffices to show that every element of the matrix $M^{-1}\mathbf{\Lambda}^{-1} = (\mathbf{B}^T\mathbf{B}+\lambda^*\mathbf{D}^T\mathbf{D})^{-1}$ is $o(1)$. Since $\mathbf{\Lambda}^{-1}$ is positive-definite, it suffices to show the diagonal elements of $M^{-1}\mathbf{\Lambda}^{-1}$ are $o(1)$. For $1 \le i \le c$, the largest eigenvalue of $M^{-1}\mathbf{\Lambda}^{-1}$ is smaller than the largest eigenvalue of $(\mathbf{B}^T\mathbf{B})^{-1}$ since $\mathbf{D}^T\mathbf{D}$ is positive semi-definite. By Lemma 2 in Zhou et al. (1998), the eigenvalues of $(\mathbf{B}^T\mathbf{B})^{-1}$ are $O(K/n)$. Hence the diagonal elements of $M^{-1}\mathbf{\Lambda}$ are all $O(K/n) = o(1)$.

Lemma A.2 *The B-splines satisfy* $\sum_{k=1}^{K+p} B_k(x) = 1$ *for any* $x \in (0,1)$.

See page 201 in de Boor (1978).

Lemma A.3 *The B-splines with degree at least* 1 *satisfy* $\sum_{k=1}^{K+p} B_k(x)\{Kx - k + (p+1)/2\} = 0$ *for any* $x \in (0,1)$.

Proof of Lemma A.3: By Lemma A.2, $\sum_{k=1}^{K+p} B_k(x)\{Kx - k + (p+1)/2\} = 0$ is equivalent to

$$\sum_{k=1}^{K+p} B_k(x)k = Kx + (p+1)/2. \tag{A.65}$$

We shall prove (A.65) by induction on p. Assume $p = 1$. Let k_x be the integer such that $x \in [k/K, (k+1)/K)$. Then $B_{k_x+1}(x) = -Kx+k+1$ and $B_{k_x+2}(x) = Kx-k$. It follows that

$$\begin{aligned}\sum_{k=1}^{K+1} B_k(x)k =& (-Kx + k_x + 1)(k_x + 1) + (Kx - k_x)(k_x + 2)\\ =& (Kx - k_x)(k_x + 2 - k_x - 1) + (k_x + 1)\\ =& Kx + 1.\end{aligned}$$

Assume now the degree of the B-splines is p. We use $B_k^{[p]}(x)$ to denote the B-splines is of degree p. We use the recursive relation of de Boor,

$$\begin{aligned} B_k^{[p]}(x) =& \frac{K}{p}\left[\left(x - \frac{k-p-1}{K}\right) B_{k-1}^{[p-1]}(x) + \left(\frac{k}{K} - x\right) B_k^{[p-1]}(x)\right] \\ =& \frac{1}{p}\left[(Kx - k + p + 1)\, B_{k-1}^{[p-1]}(x) - (Kx - k)\, B_k^{[p-1]}(x)\right]. \end{aligned} \tag{A.66}$$

It follows that

$$\begin{aligned} & p\left\{\sum_{k=1}^{K+p} B_k^{[p]}(x)k\right\} \\ =& \sum_{k=1}^{K+p}\left[(Kx - k + p + 1)\, B_{k-1}^{[p-1]}(x) - (Kx - k)\, B_{k+1}^{[p-1]}(x)\right] k \\ =& \sum_{k=1}^{K+p-1} B_{k-1}^{[p-1]}(x)(Kx - k + p + 1)k - \sum_{k=1}^{K+p-1} B_k^{[p-1]}(x)(Kx - k)k \\ =& \sum_{k=1}^{K+p-1} B_k^{[p-1]}(x)(Kx - k + p)(k+1) - \frac{1}{p}\sum_{k=1}^{K+p-1} B_k^{[p-1]}(x)(Kx - k)k \\ =& \sum_{k=1}^{K+p-1} B_k^{[p-1]}(x)(Kx - k + p + pk) \\ =& Kx + p + (p-1)\sum_{k=1}^{K+p-1} B_k^{[p-1]}(x)k \\ =& \{Kx + p + (p-1)(Kx + p/2)\} \\ =& p\{Kx + (p+1)/2\}, \end{aligned}$$

which is (A.65). Therefore, Lemma A.3 is proved.

Lemma A.4 *Let $M = n/K$ be an integer. Let $\{B_1(x), \ldots, B_c(x)\}$, where $c = K + p$, be the the B-splines basis with knots $\{-p/K, -(p-1)/K, \ldots, 0/K, 1/K, \ldots, K/K\}$. Then for $k = q+1, \ldots, K$,*

$$\sum_{i=1}^{n} B_k(x_i) = M$$

Proof of Lemma A.4: Proof by induction on p. Consider $p = 0$. $B_k(x) = 1$ if $x \in [k/K, (k+1)/K)$ and is 0 otherwise. So for fixed k, $B_k(x_i) = 1$ if and only if $(i - 1/2)/n \in [k/K, (k+1)/K)$, i.e., if and only if $i = nk/K + 1, nk/K + 1, \ldots, n(k+1)/K$. Hence the case $p = 0$ is proved. Now consider $p \geq 1$. By the recursive relation of de Boor in (A.66),

$$\begin{aligned}
\sum_i B_k^{[p]}(x_i) =& \sum_i \frac{1}{p}\left[(Kx_i - k + p + 1)B_{k-1}^{[p-1]}(x_r) - (Kx_i - k)B_k^{[p-1]}(x_i)\right] \\
=& \frac{M(-k+p+1+k)}{p} + \frac{K}{p}\sum_i x_i\left\{B_{k-1}^{[p-1]}(x_i) - B_k^{[p]}(x_i)\right\} \\
=& \frac{M(p+1)}{p} + \frac{K}{p}\left\{\sum_{i=1}^{n} x_i B_{k-1}^{[p-1]}(x_i) - \sum_{i=1}^{n-M}(x_i + 1/K)B_{k-1}^{[p-1]}(x_i)\right\} \\
=& \frac{M(p+1)}{p} + \frac{K}{p}\left\{\sum_{i=1}^{n} x_i B_{k-1}^{[p-1]}(x_i) - \sum_{r=1}^{n}(x_i + 1/K)B_{k-1}^{[p-1]}(x_i)\right\} \\
& \frac{M(p+1)}{p} + \frac{1}{p}\sum_{i=1}^{n} B_{k-1}^{[p-1]}(x_i) \\
=& \frac{M(p+1)}{p} - \frac{1}{p}M \\
=& M.
\end{aligned}$$

So Lemma A.4 is proved.

Lemma A.5 $P(1) = 1, P'(1) = p$.

Proof of Lemma A.5: The expression of $P(x)$ in (A.9) is rewritten here,

$$P(x) = u_p + u_{p-1}x + \cdots + u_0 x^p + u_1 x^{p+1} + \cdots + u_p x^{2p}.$$

Hence, $P(1) = 2\sum_{i=1}^{p} u_i + u_0$ and $P'(1) = p(2\sum_{i=1}^{p} u_i + u_0)$, so we only need to show that $2\sum_{i=1}^{p} u_i + u_0 = 1$. Let $\mathbf{C} = \mathbf{B}^T\mathbf{B}/M$. By (A.10), if $p < i < c - p$, then the coefficient vector $(u_p, u_{p-1}, \cdots, u_0, u_1, \cdots, u_p)^T$ equals $(C_{i,i-p}, C_{i,i-p+1}, \cdots, C_{i,i}, C_{i,i+1}, \cdots, C_{i,i+p})^T$. Thus, $2\sum_{i=1}^{p} u_i + u_0 =$

$\sum_{|i-j|\le p} C_{i,j} = \sum_j C_{i,j}$ because $C_{i,j} = 0$ if $|i-j| > p$. Since $C_{i,j} = \sum_r B_i(x_r)B_j(x_r)/M$, $2\sum_{i=1}^p u_i + u_0 = \sum_r \{B_i(x_r)\sum_j B_j(x_r)\}/M = \sum_r B_i(x_r)/M = 1$, where the last equality holds by Lemma A.4.

Lemma A.6 *If $\{\psi_1, \ldots, \psi_m\}$ are the m roots of $x^{2m} + (-1)^m = 0$ satisfying that the real part of ψ_ν is positive, then*

$$\prod_{j\neq\nu}(\psi_\nu^2 - \psi_j^2) = (-1)^{m+1} m \psi_\nu^{-2}. \tag{A.67}$$

Proof of Lemma A.6: It is easy to see that $\{\psi_1^2, \ldots, \psi_m^2\}$ are the m roots of $x^m + (-1)^m = 0$. Thus, $\prod_{j=1}^m (x - \psi_j^2) = (-1)^m$. Taking derivative of $\prod_{j=1}^m (x-\psi_j^2)$ with respect to x and letting $x = \psi_\nu^2$ give (A.67).

Lemma A.7 *Suppose $g(x) = \exp(-b|x|)$ with $b \neq 0$.*

$$\sum_{k,r} B_k(x)B_r(x_i)g(\frac{\bar{x}_k - \bar{x}_r}{h_n}) = \left\{1 - \frac{b}{Kh_n}\tilde{g}(x, x_i) + O\{(Kh_n)^{-2}\}\right\} g(\frac{x - x_i}{h_n}),$$

where

$$\tilde{g}(x, x_i) = \begin{cases} 2\sum_{k<r} B_k(x)B_r(x_i)(r-k) & \text{if} \quad x \ge x_i, \\ 2\sum_{k>r} B_k(x)B_r(x_i)(k-r) & \text{if} \quad x < x_i. \end{cases} \tag{A.68}$$

Proof of Lemma A.7: Suppose that $x \ge x_i$. Take a Taylor expansion of $g(x)$ at the point $\frac{x-x_i}{h_n}$,

$$\begin{aligned} g(\frac{\bar{x}_k - \bar{x}_r}{h_n}) &= g(\frac{x - x_i}{h_n}) \left\{1 - \frac{b}{h_n}(|\bar{x}_k - \bar{x}_r| - |x - x_i|) + O\{(Kh_n)^{-2}\}\right\} \\ &= g(\frac{x - x_i}{h_n}) \left\{1 - \frac{b}{Kh_n}(|k - r| - Kx + Kx_i) + O\{(Kh_n)^{-2}\}\right\}. \end{aligned}$$

Hence if we drop the term $g(\frac{x-x_i}{h_n})O\{(Kh_n)^{-2}\}$ in the above equality,

$$
\begin{aligned}
&\sum_{k,r} B_k(x)B_r(x_i)g(\frac{\bar{x}_k - \bar{x}_r}{h_n}) \\
&= g(\frac{x-x_i}{h_n}) \sum_{k,r} B_k(x)B_r(x_i) \left\{ 1 - \frac{b}{Kh_n}(|k-r| - Kx + Kx_i) \right\} \\
&= g(\frac{x-x_i}{h_n}) \left\{ 1 - \frac{b}{Kh_n} \sum_{k,r} B_k(x)B_r(x_i)(|k-r| - Kx + Kx_i) \right\} \\
&= g(\frac{x-x_i}{h_n}) \left\{ 1 - \frac{b}{Kh_n} \sum_{k,r} B_k(x)B_r(x_i) \left(|k-r| - k + \frac{p+1}{2} + Kx_i \right) \right\} \\
&= g(\frac{x-x_i}{h_n}) \left\{ 1 - \frac{b}{Kh_n} \sum_{k,r} B_k(x)B_r(x_i)(|k-r| + r - k) \right\} \\
&= g(\frac{x-x_i}{h_n}) \left\{ 1 - \frac{2b}{Kh_n} \sum_{k<r} B_k(x)B_r(x_i)(r-k) \right\}.
\end{aligned}
$$

Note that in the above derivation, we used Lemma A.2 and A.3. The other case when $x < x_i$ can be similarly proved.

Lemma A.8 *The function $\tilde{g}$ defined in (A.68) satisfies*

$$\tilde{g}(x, x_i) = 0 \quad \text{if } |x - x_i| \geq (p+1)/K.$$

Proof of Lemma A.8: Suppose $x \geq x_i$. When $x - x_i \geq (p+1)/K$ and $k < r$, either $B_k(x)$ or $B_r(x_i)$ will be 0. The other case can be similarly proved.

Lemma A.9 *Suppose $g(x) = \exp(-b|x|)$ with $b \neq 0$.*

$$\sum_r B_r(x_i) g(\frac{r}{Kh_n}) = \left[1 + O\{(Kh_n)^{-1}\}\right] g(\frac{x_i}{h_n}).$$

Proof of Lemma A.9: Take a Taylor expansion of $g(x)$ at the point $\frac{x_i}{h_n}$,

$$\begin{aligned} g(\frac{r}{Kh_n}) &= g(\frac{x_i}{h_n})\left\{1 - \frac{b}{h_n}\left(\frac{r}{K} - x_i\right) + O\{(Kh_n)^{-1}\}\right\} \\ &= g(\frac{x_i}{h_n})\left\{1 - \frac{b}{Kh_n}(r - Kx_i) + O\{(Kh_n)^{-1}\}\right\}. \end{aligned}$$

Hence if we drop the term $g(\frac{x_i}{h_n})O\{(Kh_n)^{-1}\}$ in the above equality,

$$\begin{aligned} \sum_r B_r(x_i)g(\frac{r}{Kh_n}) &= g(\frac{x_i}{h_n})\sum_r B_r(x_i)\left\{1 - \frac{b}{Kh_n}(r - Kx_i)\right\} \\ &= g(\frac{x_i}{h_n})\left\{1 - \frac{b}{Kh_n}\sum_r B_r(x_i)(r - Kx_i)\right\} \\ &= g(\frac{x_i}{h_n})\left\{1 - \frac{b}{Kh_n}\sum_r B_r(x_i)\frac{p+1}{2}\right\} \\ &= g(\frac{x_i}{h_n})\left(1 - \frac{p+1}{Kh_n}\right). \end{aligned}$$

Lemma A.10 *Assume ψ is a complex number and $|\psi| = 1$. For any nonnegative integer ℓ,*

$$\int x^\ell e^{-\psi x}\mathrm{d}x = -e^{-\psi x}\sum_{k=1}^{\ell+1}\frac{\ell! x^{\ell-k+1}}{(\ell-k+1)!}\bar{\psi}^k,$$

where $\bar{\psi}$ is the conjugate of ψ.

Proof of Lemma A.10: The results of indefinite integrals of $\int x^\ell e^{ax}\cos(bx)\mathrm{d}x$ and $\int x^\ell e^{ax}\sin(bx)\mathrm{d}x$ are given by results 3 and 4 on page 230 of Gradshteyn and Ryzhik (2007).

Lemma A.11 *Assume $|\psi| = 1$ with positive real part. For any nonnegative integer ℓ,*

$$\int_0^\infty x^\ell e^{-\psi x}\mathrm{d}x = \ell!\bar{\psi}^{\ell+1},$$

where $\bar{\psi}$ is the conjugate of ψ.

Proof of Lemma A.11: See Lemma A.10.

Lemma A.12 *If ℓ is even and $2 \leq \ell \leq 2m-2$,*

$$\sum_{\nu=1}^{m} \psi_\nu^\ell = 0.$$

Proof of Lemma A.12: Assume $\{z_1, z_2, \ldots, z_{2m}\}$ are all the roots of the equation $x^{2m}+(-1)^m = 0$. Since ℓ is even, we can show that $\sum_{\nu=1}^{m} \psi_\nu^\ell = 1/2 \sum_{i=1}^{2m} z_i^\ell$ because if $a + b\imath$ is a root of $x^{2m} + (-1)^m = 0$, then $\pm a \pm b\imath$ are also roots. Assume m is odd first. Let $\omega = e^{\imath\pi/m}$. Note that ω is a primitive root of $x^{2m} = 1$, and we can organize $\{z_1, \ldots, z_{2m}\}$ in such a way that $z_i = \omega^i$. It follows that

$$\sum_{i=1}^{2m} z_i^\ell = \sum_{i=1}^{2m} \omega^{\ell i} = \omega^\ell \frac{1-\omega^{2m\ell}}{1-\omega^\ell} = 0.$$

For the case m is even, let $\omega_0 = e^{\imath\pi/(2m)}$. We can also write $z_i = \omega_0^{1+2i}$, then

$$\sum_{i=1}^{2m} z_i^\ell = \sum_{i=1}^{2m} \omega_0^{\ell(1+2i)} = \omega_0^\ell \frac{1-\omega_0^{4m\ell}}{1-\omega_0^{2\ell}} = 0.$$

Lemma A.13

$$\int_{-\infty}^{\infty} x^\ell H_m(x) \mathrm{d}x = \begin{cases} 1 & : \quad \ell = 0 \\ 0 & : \quad \ell \text{ is odd} \\ 0 & : \quad \ell \text{ is even and } 2 \leq \ell \leq 2m-2 \\ (-1)^{m+1}(2m)! & : \quad \ell = 2m \end{cases}$$

Proof of Lemma A.13: Since $H_m(x)$ is symmetric about 0, the result for odd ℓ is obvious. Assume ℓ is even. By Lemma A.11,

$$\begin{aligned}\int_{-\infty}^{\infty} x^\ell H_m(x)\mathrm{d}x &= \frac{1}{m}\sum_{\nu=1}^{m}\psi_\nu \int_0^\infty x^\ell e^{-\psi_\nu x}\mathrm{d}x \\ &= \frac{\ell!}{m}\sum_{\nu=1}^{m}\psi_\nu \bar{\psi}_\nu^{\ell+1} \\ &= \frac{(-1)^{m+1}\ell!}{m}\sum_{\nu=1}^{m}\psi_\nu^{2m-\ell}.\end{aligned}$$

If $\ell = 0$, $\int_{-\infty}^{\infty} H_m(x)\mathrm{d}x = \frac{(-1)^{m+1}}{m}\sum_{\nu=1}^{m}\psi_\nu^{2m} = 1$ as desired. If $\ell = 2m$, $\int_{-\infty}^{\infty} x^{2m}H_m(x)\mathrm{d}x = (-1)^{m+1}(2m)!$ also as desired. The case when ℓ is even and $2 \le \ell \le 2m-2$ is proved by Lemma A.12.

APPENDIX B

CODE FOR THE SANDWICH SMOOTHER

```
## pre-calculations
Ytilde = t(A1)%*%Y%*%A2
Y_square = sum(Y^2)
ytilde = as.vector(Ytilde)

## the function calculates the GCV of the sandwich smoother
fbps_gcv = function(x){

lambda = exp(x)
## two lambda's are the same
if(length(lambda)==1)
{
lambda1 = lambda
lambda2 = lambda
}
## two lambda's are different
if(length(lambda)==2){
 lambda1=lambda[1]
 lambda2=lambda[2]
}

sigma = kronecker(1/(1+lambda2*s2),1/(1+lambda1*s1))
sigma.2 = sqrt(sigma)
```

```
gcv = Y_square + sum((ytilde*sigma)^2) - 2*sum((ytilde*sigma.2)^2)
trace = sum(1/(1+lambda1*s1))*sum(1/(1+lambda2*s2))
gcv = gcv/(1-trace/(n1*n2))^2
return(gcv)
}

## the function calculates estimates of the sandwich smoother
##  with fixed smoothing parameters
fbps_est = function(x){

lambda = exp(x)
## two lambda's are the same
if(length(lambda)==1)
{
lambda1 = lambda
lambda2 = lambda
}
## two lambda's are different
if(length(lambda)==2){
 lambda1=lambda[1]
 lambda2=lambda[2]
}

sigma = kronecker(1/(1+lambda2*s2),1/(1+lambda1*s1))
sigma.2 = sqrt(sigma)
```

```
gcv = Y_square + sum((ytilde*sigma)^2) - 2*sum((ytilde*sigma.2)^2)
trace = sum(1/(1+lambda1*s1))*sum(1/(1+lambda2*s2))
gcv = gcv/(1-trace/(n1*n2))^2

Theta = Sigi1_sqrt%*%U1%*%diag(1/(1+lambda1*s1))%*%Ytilde
Theta = Theta%*%diag(1/(1+lambda2*s2))%*%t(U2)%*%Sigi2_sqrt
hatY = B1%*%Theta%*%t(B2)
result=list(lambda = c(lambda1,lambda2), hatY = hatY,
                   trace = trace, gcv = gcv, Theta = Theta)
return(result)
}
```

APPENDIX C

A GLAM ALGORITHM FOR THE E-M ESTIMATOR

We shall continue using the notation in Chapter 2. We define

$$\mathbf{W} = (\mathbf{B}_2^T\mathbf{B}_2) \otimes (\mathbf{B}_1^T\mathbf{B}_1) + \lambda_1\mathbf{I}_{c_2} \otimes \mathbf{D}_1^T\mathbf{D}_1 + \lambda_2\mathbf{D}_2^T\mathbf{D}_2 \otimes \mathbf{I}_{c_1}.$$

Then the vector of fitted values for E-M/GLAM is

$$\hat{\mathbf{y}} = (\mathbf{B}_2 \otimes \mathbf{B}_1)\mathbf{W}^{-1}(\mathbf{B}_2 \otimes \mathbf{B}_1)^T\mathbf{y}.$$

Using the notation $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{U}_1, \mathbf{U}_2$, and $\mathbf{s}_1, \mathbf{s}_2$ in Section 2.2.2 of Chapter 2, the above becomes

$$\hat{\mathbf{y}} = (\mathbf{A}_2 \otimes \mathbf{A}_1)\widetilde{\mathbf{W}}^{-1}(\mathbf{A}_2 \otimes \mathbf{A}_1)^T\mathbf{y}, \tag{C.1}$$

where

$$\widetilde{\mathbf{W}} = \mathbf{I}_{c_2c_1} + \lambda_1\mathbf{U}_2^T(\mathbf{B}_2^T\mathbf{B}_2)^{-1}\mathbf{U}_2 \otimes \mathrm{diag}(\mathbf{s}_1) + \lambda_2\mathrm{diag}(\mathbf{s}_2) \otimes \mathbf{U}_1^T(\mathbf{B}_1^T\mathbf{B}_1)^{-1}\mathbf{U}_1.$$

We first let $\widetilde{\mathbf{Y}} = \mathbf{A}_1^T\mathbf{Y}\mathbf{A}_2$ and $\tilde{\mathbf{y}} = vec(\widetilde{\mathbf{Y}})$. Then $\tilde{\mathbf{y}} = (\mathbf{A}_2 \otimes \mathbf{A}_1)^T\mathbf{y}$. Next we define $\boldsymbol{\gamma} = \widetilde{\mathbf{W}}^{-1}\tilde{\mathbf{y}}$, a c_1c_2 vector. Then we have $\hat{\mathbf{y}} = (\mathbf{A}_2 \otimes \mathbf{A}_1)\boldsymbol{\gamma}$. The relation between $\boldsymbol{\gamma}$ and the coefficients vector $\hat{\boldsymbol{\theta}}$ is: $\hat{\boldsymbol{\theta}} = \left\{(\mathbf{B}_2^T\mathbf{B}_2)^{-1/2}\mathbf{U}_2 \otimes (\mathbf{B}_1^T\mathbf{B}_1)^{-1/2}\mathbf{U}_1)\right\}\boldsymbol{\gamma}$. Hence if we let $\boldsymbol{\Gamma}$ be the $c_1 \times c_2$ matrix such that $vec(\boldsymbol{\Gamma}) = \boldsymbol{\gamma}$, then

$$\widehat{\boldsymbol{\Theta}} = \left\{(\mathbf{B}_1^T\mathbf{B}_1)^{-1/2}\mathbf{U}_1\right\}\boldsymbol{\Gamma}\left\{(\mathbf{B}_2^T\mathbf{B}_2)^{-1/2}\mathbf{U}_2\right\}^T. \tag{C.2}$$

We use the generalized cross validation for selecting the smoothing parameters. First we derive that

$$\|\widehat{\mathbf{Y}} - \mathbf{Y}\|_F^2 = \hat{\mathbf{y}}^T\hat{\mathbf{y}} - 2\hat{\mathbf{y}}^T\mathbf{y} + \mathbf{y}^T\mathbf{y}.$$

Because

$$\begin{aligned}\hat{\mathbf{y}}^T\hat{\mathbf{y}} &= \boldsymbol{\gamma}^T(\mathbf{A}_2 \otimes \mathbf{A}_1)^T(\mathbf{A}_2 \otimes \mathbf{A}_1)\boldsymbol{\gamma} \\ &= \boldsymbol{\gamma}^T(\mathbf{A}_2^T\mathbf{A}_2 \times \mathbf{A}_1^T\mathbf{A}_1)\boldsymbol{\gamma} \\ &= \boldsymbol{\gamma}^T\boldsymbol{\gamma}.\end{aligned}$$

and similarly

$$\hat{\mathbf{y}}^T\mathbf{y} = \boldsymbol{\gamma}^T(\mathbf{A}_2 \otimes \mathbf{A}_1)^T\mathbf{y} = \boldsymbol{\gamma}^T\tilde{\mathbf{y}},$$

we obtain that

$$\|\widehat{\mathbf{Y}} - \mathbf{Y}\|_F^2 = \boldsymbol{\gamma}^T\boldsymbol{\gamma} - 2\boldsymbol{\gamma}^T\tilde{\mathbf{y}} + \mathbf{y}^T\mathbf{y}. \tag{C.3}$$

For the trace of smoother matrix, by (C.1),

$$tr\left\{(\mathbf{A}_2 \otimes \mathbf{A}_1)\widetilde{\mathbf{W}}^{-1}(\mathbf{A}_2 \otimes \mathbf{A}_1)^T\right\} = \mathrm{tr}\left(\widetilde{\mathbf{W}}^{-1}\right). \tag{C.4}$$

With (C.2), (C.3) and (C.4) we obtain a GLAM algorithm for the E-M estimator as in Algorithm 1.

Input : $\mathbf{Y}$, $\mathbf{B}_1, \mathbf{B}_2$, $\mathbf{D}_1, \mathbf{D}_2$, c_1, c_2, n_1, n_2, and a list of pairs of smoothing parameters
Output: $\widehat{\boldsymbol{\Theta}}$ of dimension $c_1 \times c_2$ and the matrix of fitted values $\widehat{\mathbf{Y}}$

begin
  Compute $\mathbf{A}_i, \mathbf{U}_i, \mathbf{s}_i, (\mathbf{B}_i^T\mathbf{B}_i)^{-1/2}\mathbf{U}_i, \mathbf{U}_i^T(\mathbf{B}_i^T\mathbf{B}_i)^{-1}\mathbf{U}_i, i = 1, 2$;
  $\mathbf{P}_1 = \mathbf{U}_2^T(\mathbf{B}_2^T\mathbf{B}_2)^{-1}\mathbf{U}_2 \otimes \mathrm{diag}(\mathbf{s}_1)$;
  $\mathbf{P}_2 = \mathrm{diag}(\mathbf{s}_2) \otimes \mathbf{U}_1^T(\mathbf{B}_1^T\mathbf{B}_1)^{-1}\mathbf{U}_1$;
  $\widetilde{\mathbf{Y}} = \mathbf{A}_1^T\mathbf{Y}\mathbf{A}_2, \tilde{\mathbf{y}} = vec(\widetilde{\mathbf{Y}})$;
  $s_y = \mathbf{y}^T\mathbf{y}$;
  for *every pair of smoothing parameters* (λ_1, λ_2) **do**
    $\widetilde{\mathbf{W}} = \mathbf{I}_{c_2c_1} + \lambda_1\mathbf{P}_1 + \lambda_2\mathbf{P}_2$;
    $\boldsymbol{\gamma} = \widetilde{\mathbf{W}}^{-1}\tilde{\mathbf{y}}$;
    $GCV = \boldsymbol{\gamma}^T\boldsymbol{\gamma} - 2\boldsymbol{\gamma}^T\tilde{\mathbf{y}} + s_y$;
    $trace = \mathrm{tr}\left(\widetilde{\mathbf{W}}^{-1}\right)$;
    $GCV = GCV/\left\{1 - trace/(n_1n_2)\right\}^2$;
  end
  select the pair $(\lambda_1^*, \lambda_2^*)$ *that has the smallest GCV*;
  $\widetilde{\mathbf{W}} = \mathbf{I}_{c_2c_1} + \lambda_1^*\mathbf{P}_1 + \lambda_2^*\mathbf{P}_2$;
  $\boldsymbol{\gamma} = \widetilde{\mathbf{W}}^{-1}\tilde{\mathbf{y}}$ and define $\boldsymbol{\Gamma}$, a $c_1 \times c_2$ matrix, such that $vec(\boldsymbol{\Gamma}) = \boldsymbol{\gamma}$;
  $\widehat{\boldsymbol{\Theta}} = \left\{(\mathbf{B}_1^T\mathbf{B}_1)^{-1/2}\mathbf{U}_1\right\}\boldsymbol{\Gamma}\left\{(\mathbf{B}_2^T\mathbf{B}_2)^{-1/2}\mathbf{U}_2\right\}^T$;
  $\widehat{\mathbf{Y}} = \mathbf{A}_1\widetilde{\mathbf{Y}}\mathbf{A}_2^T$;
end

Algorithm 1: Algorithm for E-M/GLAM

From Algorithm 1, we see that most of the computation time is from computing $\widetilde{\mathbf{W}}^{-1}\tilde{\mathbf{y}}$, which requires about $c_1^3 c_2^3$ computations and is re-calculated for every pair of smoothing parameters.

APPENDIX D

CODE FOR FAST COVARIANCE FUNCTION ESTIMATION

```
## pre-calculations
Ytilde = t(AS)%*%Y
C_diag = rowSums(Ytilde^2)

Y_square = sum(Y^2)
Ytilde_square = sum(Ytilde^2)

## the function computes the GCV for FACE
face_gcv = function(x){

    lambda = exp(x)
    lambda_s = (lambda*s)^2/(1 + lambda*s)^2
    gcv = sum(C_diag*lambda_s) - Ytilde_square + Y_square
    trace = sum(1/(1 + lambda*s))^2
    gcv = gcv/(1 - trace/m)^2
    return(gcv)
  }

## the function computes the eigenfunctions and eigenvalues
## from FACE with a fixed smoothing parameter
face_est = function(x){

lambda = exp(x)
SigmaS = 1/(1 + lambda*s)%x%t(1/(1+lambda*s))
```

```
temp = n^(-1)*(C*SigmaS)
Eigen = eigen(temp)
A = Eigen$vectors
d = Eigen$values/m
result = list(AS = AS, A = A, d = d)
}

## the following code computes the principal scores using BLUPS

A.N = AS%*%A[,1:N]
d = d[1:N]
sigma_square = Y_square/(m*n) - sum(d)
Sigma = m^(-1/2)* diag(d/(d+sigma_square))
Scores = (Sigma%*%t(A.N))%*%Ytilde
```

BIBLIOGRAPHY

Baik, J. and Silverstein, J.W. (2005), "Eigenvalues of large sample covariance matrices of spiked population models," *J. Multivar. Anal.*, 97, 1382-1408.

Besse, P., Cardot, H. and Ferraty, F. (1997), "Simultaneous nonparametric regressions of unbalanced longitudinal data," *Comput. Statist. Data Anal.*, 24, 255-270.

Besse, P. and Ramsay, J. O. (1986), "Principal components analysis of sampled functions," *Psychometrika*, 51, 285-311.

Capra, W.B. and Müller, H.G. (1997), "An accelerated-time model for response curves," *J. Amer. Statist. Assoc.*, 92, 72-83.

Cardot, H. (2000), "Nonparametric estimation of smoothed principal components analysis of sampled noisy functions," *J. Nonparametr. Statist.* 12, 503-538.

Claeskens, G., Krivobokova, T., and Opsomer, J. D. (2009), "Asymptotic properties of penalized spline estimators," *Biometrika*, 96, 529-544.

Crainiceanu, C., Staicu, A., and Di, C. (2010), "Generalized multilevel functional regression," *J. Amer. Statist. Assoc.*, 104(488), 1550-1561.

Crainiceanu, C., Staicu, A., Ray, S. and Punjabi, N. (2010), "Bootstrap-based inference on the difference in the means of two correlated functional processes," available at `http://biostats.bepress.com/jhubiostat/paper225/`.

Currie, I.D., Durban, M. and Eilers, P.H.C. (2006), "Generalized linear array models with applications to multidimensional smoothing," *J. R. Statist. Soc. Ser. B*, 68, 259-280.

de Boor, C. (1978), *A Practical Guide to Splines*, Berlin: Springer.

Di, C., Crainiceanu, C. M., Caffo, B.S., and Punjabi, N. (2009), "Multilevel functional principal component analysis," *Ann. Appl. Statisti.*, 3, 458-488.

Dierckx, P. (1982), "A fast algorithm for smoothing data on a rectangular grid while using spline functions," *SIAM J. Numer. Anal.*, 19, 1286-1304.

Dierckx, P. (1995), *Curve and Surface Fitting with Splines*, Clarendon Press, Oxford.

Diggle, P. J., Liang, K.Y. and Zeger, S. L. (1994), *Analysis of Longitudinal Data*, New York: Oxford University Press.

Durrett, R. (2005), *Probability: Theory and Examples*, Third Edition, Thomson.

Eilers, P.H.C., Currie, I.D. and Durban M. (2006), "Fast and compact smoothing on large multidimensional grids," *Comput. Statist. Data Anal.*, 50, 61-76.

Eilers, P.H.C. and Goeman, J.J. (2004), "Enhancing scatterplots with smoothed densities," *Bioinformatics*, 20, 623-628.

Eilers, P.H.C. and Marx, B.D. (1996), "Flexbile smoothing with B-splines and penalties (with Discussion)," *Statist. Sci.*, 11, 89-121.

Eilers, P.H.C. and Marx, B.D. (2003), "Mulitvariate calibration with temperature interaction using two-dimensional penalized signal regression," *Chemometrics and Intelligent Laboratory Systems*, 66, 159-174.

Fan, J. (1992), "Design-adaptive nonparametric regression," *J. Amer. Statist. Assoc.*, 87, 998-1004.

Ferraty, F. and Vieu, P. (2006), *Nonparametric Functional Data Analysis: Methods, Theory, Applications and Implementations*, New York: Springer.

Gasser, T. and Müller, H.G. (1979), "Kernel Estimation of Regression Functions," in *Smoothing Techniques for Curve Estimation*, Lecture Notes in Mathematics 757, eds. T. Gasser and M. Rosenblatt, Heidelberg: Springer-Verlag, pp. 23-68.

Goldsmith, J., Bobb, J., Crainiceanu, C.M., Caffo, B., and Reich, D. (2011), "Penalized functional regression," *J. Comput. Graph. Statist.*, 20, 830-851.

Goldsmith, J., Crainiceanu, C., Caffo, B., and Reich, D. (2012), " Longitudinal penalized functional regression for cognitive outcomes on neuronal tract measurements," *J. R. Statist. Soc. Ser. C*, to appear.

Gradshteyn, I.S. and Ryzhik, I.M.(2007), *Table of Integrals, Series, and Products*, New York: Academic Press.

Greven, S., Crainiceanu, C., Caffo, B. and Reich, D. (2010), "Longitudinal functional principal component," *Electronic J. Statist.*, 4, 1022-1054.

Gu, C. (2002), *Smoothing Spline ANOVA Models*, New York: Springer.

Hall, P., Müller, H.G., and Wang, J.L. (2006), "Properties of principal component methods for functional and longitudinal data analysis," *Ann. Statist.*, 34, 1493-1517.

Hansen, B.E. (2008), "Uniform convergence rates for kernel estimation with dependent data," *Econometric Theory*, 24, 726-748.

Hastie, T. and Tibshirani, R.(1993), "Varying-Coefficients Models," *J. R. Statist. Soc. Ser. B*, 55, 757-796.

Holst, U., Hössjer, O., Björklund, C., Ragnarson, P. and Edner, H. (1996), "Locally weighted least squares kernel regression and statistical evaluation of LIDAR measurements," *Environmetrics* 7: 401-416.

Hutchinson, M.F. and de Hoog, F.R. (1985), "Smoothing noisy data with spline functions," *Numer. Math.*, 47, 99-106.

Laub, A.J.(2005), *Matrix Analysis for Scientists and Engineers*, SIAM.

Johnstone, I.M. (2001), "On the distribution of the largest principal component," *Ann. Statist.*, 29, 295-327.

Johnstone, I.M. and Lu, A.Y. (2009), "On consistency and sparsity for principal components analysis in high dimensions," *J. Amer. Statist. Assoc.*, 29, 295-327.

Karhunen, K. (1947), "Uber lineare methoden in der wahrscheinlichkeitsrechnung," *Annales Academie Scientiarum Fennicae*, 37, 1-79.

Kauermann, G., Krivobokova, T. and Fahrmeir, L. (2009), "Some asymptotic results on generalized penalized spline smoothing," *J. R. Statist. Soc. Ser. B*, 71, 487-503.

Li,Y. and Ruppert D. (2008), "On the asymptotics of penalized splines," *Biometrika*, 95, 415-436.

Marx, B.D. and Eilers, P.H.C. (2005), "Multdimensional Penalized Signal Regression," *Technometrics*, 47, 13-22.

Messer, K. and Goldstein, L.(1993), "A new class of kernels for nonparametric curve estimation," *Ann. Statist.*, 21, 179-195.

Nadaraya, E. A. (1964), "On Estimating Regression," *Theory of Probability and Its Applications*, 9, 141-142.

Opsomer, J.D. and Hall, P. (2005), "Theory for penalised spline regression," *Biometrika*, 95, 417-436.

O'Sullivan, F. (1986), "A statistical perspective on ill-posed inverse problems (with discussion)," *Statist. Sci.*, 1, 505-527.

Ramsay, J. O. and Silverman, B. W. (2002), *Applied Functional Data Analysis: Methods and Case Studies*, New York: Springer.

Ramsay, J. O. and Silverman, B. W. (2005), *Functional Data Analysis, 2nd Edition*, New York: Springer.

Ramsay, J. O. and Dalzell, C. J. (1991), "Some tools for functional data analysis (with Discussion)," *J. R. Statist. Soc. Ser. B*, 53, 539-572.

Rice, J. and Silverman, B. (1991), "Estimating the mean and covariance structure nonparametrically when the data are curves," *J. R. Statist. Soc. Ser. B*, 53, 233-243.

Ruppert, D. (2002), "Selecting the number of knots for penalized splines," *J. Comput. Graph. Statist.*, 1, 735-757.

Ruppert, D., Wand, M.P. and Carroll, R.J. (2003), *Semiparametric Regression*, Cambridge: Cambridge University Press.

Ruppert, D., Wand, M.P., Holst, U. and Hössjer, O. (1997), "Local polynomial variance function estimation," *Technometrics*, 39: 262-273.

Seber, G.A.F. (2007), *A Matrix Handbook for Statisticians*, New Jersey: Wiley-Interscience.

Silverman, B.W. (1984), "Spline smoothing: the equivalent variable kernel method," *Ann. Statist.*, 12, 898-916.

Staniswalis, J.G. and Lee, J.J. (1998), "Nonparametric regression analysis of longitudinal data," *J. Amer. Statist. Assoc.*, 93, 1403-1418.

Stein, M. (1999), *Interpolation of Spatial Data: Some Theory for Kriging*, New York: Springer.

Stone, C.J. (1980), "Optimal rates of convergence for nonparametric estimators," *Ann. Statist.*, 8, 1348-1360.

Tievsky, A. L., Ptak, T. and Farkas, J. (1999), "Investigation of apparent diffusion coefficient and diffusion tensor anisotropy in acute and chronic multiple sclerosis lesions," *Amer. J. Neuroradiology*, 20, 1491–1499.

Wahba, G. (1990), *Spline Models for Observational Data*, Philadelphia: SIAM.

Wand, M.P. and Jones, M.C. (1995), *Kernel Smoothing*, London: Chapman &Hall.

Wang X. and Shen J. (2010), "A class of grouped Brunk estimators and penalized spline estimators for monotone regression," *Biometrika*, 97, 585-601.

Wang, X., Shen, J. and Ruppert, D. (2011), "Local Asymptotics of P-spline Smoothing," *Electronic J. Statist.*, 4, 1-17.

Watson, G.S. (1964), "Smooth Regression Analysis," *Sankhya, Ser. A*, 26, 359-372.

Werring, D., Clark, C., Barker, G., Thompson, A. and Miller D. (1999), "Diffusion tensor imaging of lesions and normal-appearing white matter in multiple sclerosis," *Neurology*, 52, 1626–1632.

Wood, S.N. (2003), "Thin plate regression splines," *J. R. Statist. Soc. Ser. B*, 65, 95-114.

Wood, S.N. (2006), *Generalized additive models: an introduction with R*, London: Chapman &Hall.

Yao, F. and Lee C.M. (2006), "Penalized spline models for functional principal component analysis," *J. R. Statist. Soc. Ser. B*, 68, 3-25.

Yao, F., Müller, H., Clifford, A.J. Dueker, S.R., Follett, J., Lin, Y., Buchholz, B.A. and Vogel, J.S. (2003), "Shrinkage estimation for functional principal component scores with application to the population kinetics of plasma folate," *Biometrics*, 59, 676-685.

Zhou, S., Shen, X. and Wolfe, D.A. (1998), "Local asymptotics for regression splines and confidence regions," *Ann. Statist.*, 26, 1760-1782.

Zipunnikov, V., Caffo, B. S., Crainiceanu, C. M., Yousem D.M., Davatzikos, C., and Schwartz, B.S. (2011), "Multilevel functional principal component analysis for high-dimensional data," *J. Comput. Graph. Statist.*, 20(4), 852-873.

Zipunnikov, V., Greven, S., Caffo, B.S., and Crainiceanu, C.M. (2012), "Longitudinal high-dimensional data analysis," available at `http://biostats.bepress.com/jhubiostat/paper234/`.